IPC

*ASSOCIATION CONNECTING
ELECTRONICS INDUSTRIES*

IPC-2511B

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

IPC-2511B

January 2002

A standard developed by IPC

The Principles of Standardization

In May 1995 the IPC's Technical Activities Executive Committee adopted Principles of Standardization as a guiding principle of IPC's standardization efforts.

Standards Should:	**Standards Should Not:**
• Show relationship to Design for Manufacturability (DFM) and Design for the Environment (DFE)	• Inhibit innovation
• Minimize time to market	• Increase time-to-market
• Contain simple (simplified) language	• Keep people out
• Just include spec information	• Increase cycle time
• Focus on end product performance	• Tell you how to make something
• Include a feedback system on use and problems for future improvement	• Contain anything that cannot be defended with data

Notice

IPC Standards and Publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for his particular need. Existence of such Standards and Publications shall not in any respect preclude any member or nonmember of IPC from manufacturing or selling products not conforming to such Standards and Publication, nor shall the existence of such Standards and Publications preclude their voluntary use by those other than IPC members, whether the standard is to be used either domestically or internationally.

Recommended Standards and Publications are adopted by IPC without regard to whether their adoption may involve patents on articles, materials, or processes. By such action, IPC does not assume any liability to any patent owner, nor do they assume any obligation whatever to parties adopting the Recommended Standard or Publication. Users are also wholly responsible for protecting themselves against all claims of liabilities for patent infringement.

IPC Position Statement on Specification Revision Change

It is the position of IPC's Technical Activities Executive Committee (TAEC) that the use and implementation of IPC publications is voluntary and is part of a relationship entered into by customer and supplier. When an IPC standard/guideline is updated and a new revision is published, it is the opinion of the TAEC that the use of the new revision as part of an existing relationship is not automatic unless required by the contract. The TAEC recommends the use of the lastest revision. Adopted October 6. 1998

Why is there a charge for this standard?

Your purchase of this document contributes to the ongoing development of new and updated industry standards. Standards allow manufacturers, customers, and suppliers to understand one another better. Standards allow manufacturers greater efficiencies when they can set up their processes to meet industry standards, allowing them to offer their customers lower costs.

IPC spends hundreds of thousands of dollars annually to support IPC's volunteers in the standards development process. There are many rounds of drafts sent out for review and the committees spend hundreds of hours in review and development. IPC's staff attends and participates in committee activities, typesets and circulates document drafts, and follows all necessary procedures to qualify for ANSI approval.

IPC's membership dues have been kept low in order to allow as many companies as possible to participate. Therefore, the standards revenue is necessary to complement dues revenue. The price schedule offers a 50% discount to IPC members. If your company buys IPC standards, why not take advantage of this and the many other benefits of IPC membership as well? For more information on membership in IPC, please visit www.ipc.org or call 847/790-5372. For more information on GenCAM, please visit www.gencam.org or call 847/790-5342.

Thank you for your continued support.

GenCAM

[MANGN]

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

A standard developed by the IPC Electronic Data Transfer Generic Requirements Task Group (2-11i) of the IPC Data Generation and Transfer Committee (2-10) of IPC.

The GenCAM format is intended to provide CAD-to-CAM, or CAM-to-CAM data transfer rules and parameters related to manufacturing printed boards and printed board assemblies.

Approved American National Standard

ANSI

February 15, 2002

Users of this standard are encouraged to participate in the development of future revisions.

Contact:

IPC
2215 Sanders Road
Northbrook, Illinois
60062-6135
Tel 847 509.9700
Fax 847 509.9798

Acknowledgment

Table of Contents

Generic Requirements for Implementation of Product Manufacturing Description Data and Transfer Methodology (MANGN)

1 SCOPE

This standard specifies the XML schema that represents the data file format used to describe printed board and printed board assembly products with details sufficient for tooling, manufacturing, assembly, inspection and testing 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   GenCAM Focus

The GenCAM format requirements are provided in a series of standards focused on printed board manufacturing, assembly, inspection, and testing. This standard series consists of a generic standard (IPC-2511) which contains all the general requirements. There are seven sectional standards that are focused on the XML details necessary to accumulate information in the single GenCAM file, that addresses the needs of the manufacturing disciplines producing a particular product. The sectional standards (IPC-2512 through 2518) paraphrase the important requirements and provide suggested usage and examples for the topic covered by the sectional standard.

2 APPLICABLE DOCUMENTS

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

IPC-2512   Sectional Requirements for Implementation of Administrative Methods for Manufacturing Data Description

IPC-2513   Sectional Requirements for Implementation of Drawing Methods for Manufacturing Data Description

IPC-2514   Sectional Requirements for Implementation of Printed Board Fabrication Data Description

IPC-2515   Sectional Requirements for Implementation of Bare Board Product Electrical Testing Data Description

IPC-2516   Sectional Requirements for Implementation of Assembled Board Product Manufacturing Data Description

IPC-2517   Sectional Requirements for Implementation of Assembly In-Circuit Testing Data Description

IPC-2518   Sectional Requirements for Implementation of Part List Product Data Description

IPC-2524   PWB Fabrication Data Quality Rating System

IPC-2525   Bareboard Electrical Test Quality Rating System

IPC-2526   Printed Board Assembly Data Quality Rating System

IPC-2527   In-Circuit Test Data Quality Rating System

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 (PDX)

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

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

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. As a consequence of this change the IPC-2511 document has undergone a significant rewrite. The underlying object model of the standard has changed very little, but the examples, definitions, and organization of information have been rewritten to using the new file format and a new notation for specifying the file format. An example will help illustrate why the object model has not changed significantly.

The first example demonstrates the IPC-2511A syntax for keyword statements:

```
GROUP: "bd1";
  LINEDESC: "Line9", 1.0, SQUARE , , CENTER, 8.0, 40.0,
                TP, BOTH, 2.5, 1.3;
```

Here is the same content using the XML syntax for elements and attributes:

```
<LineDesc id = "bd1:line9" width = "1.0" lineEnd = "SQUARE"
        type = "CENTER" space = "8.0" length = "40" lineMod = "TP"
        modEnd = "BOTH" dimA = "2.5" dimB = "1.3"/>
```

This next example demonstrate the use of a `<linedesc_ref>` parameter that references the element defined in the last example:

```
RECTCENTER: 2, 4, "bd1"."line9", , , (0,0);
```
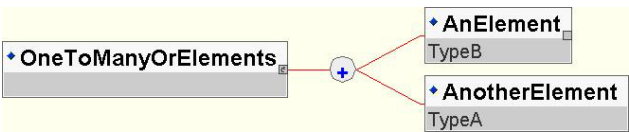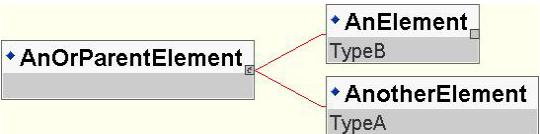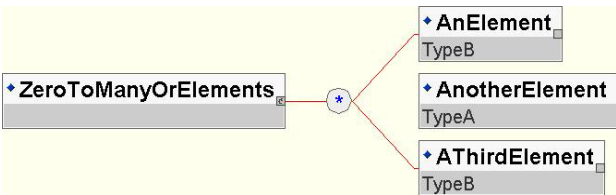
Here is the `RECTCENTER` example using the XML format of elements and attributes:

```
<RectCenter width = "4" height = "2" lineDescRef = "bd1:line9">
     <Place x = "0" y = "0"/>
</RectCenter>
```
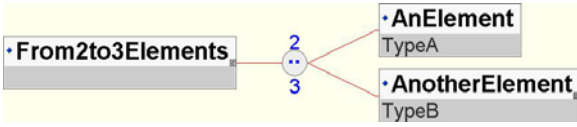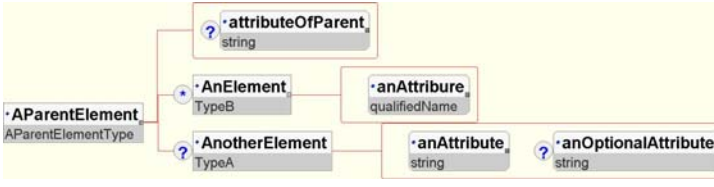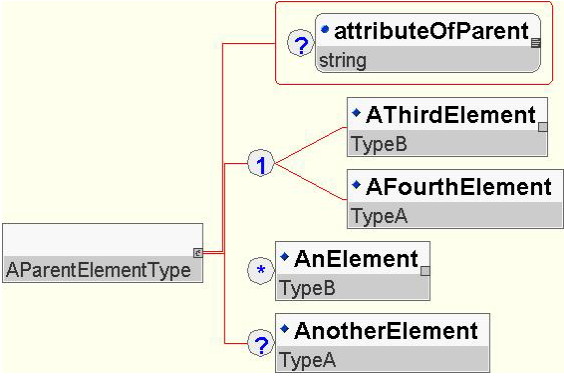
Note that the information content is unchanged. There are a few organizational change, but the primarily change is just syntactical.

In addition to the text based schema notation this document provides graphical representation of the structure of the new file format. The new diagrams are designed to effectively illustrate the structure and cardinality of elements and attributes that make up a GenCAM file. The notation in

the graphics does not provide a complete visualisation 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.. The following table provides an overview of the graphical notation used in the document.

This diagram depicts an element named `AnElement` that is of type `TypeB`. There is one attribute, named `anAttribute`, that 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 with two attributes. The attribute `anAttribute` is required. The "?" in the circle indicates that the second attribute, `anOptionalAttribute`, is optional. Both attributes are of type `string`.	

Examples:
 	
`<AnotherElement anAttribute="red" anOptionalAttribute="a string" />`
 	
`<AnotherElement anAttribute="blue" />`	

The element `OneToManyOrParentElement` is the parent of an unordered list of one or more instances of the elements `AnElement` and `AParentElement`. The "+" indicates the occurrence is one to many and the angled lines indicate this is a choice relationship ("|") between the children elements.	
< OneToManyOrParentElement>…	
The absence of an occurrence bubble declares that one and only one occurrence are allowed. The `AnOrParentElement` can have one of AnElement or AnotherElement as a child element.	
The '*' in the occurrence bubble indicates the choice is from 0 to many.	

This diagram depitcs an element, From2to3Elements. The element has no type and no attributes. It can have from 2 to 3 subelements of either AnElement or AnotherElement.	
This diagram depicts an element, `AParentElement`, of type `AParentElementType`. This element has one attribute, `attributeOfParent`, which is optional. The lines with square corners indicate that occurrences of AnElement and `AnotherElement` must appear in the order shown.	
This diagram depicts a type, `AParentElementType`, that contains a sequence starting with one of `AThirdElement` or `AFourth` element followed by `0-n AnElement` and an optional final `AnotherElement`.	

3   REQUIREMENTS

The XML schema contained in this document describes the structure of a generic computer-aided manufacturing (GenCAM) exchange format. The document specifies data elements specifically designed to establish the information exchange related to the data needed by printed board manufacturing, assembly, inspection, and test.

The GenCAM XML schema defines the configuration of mandatory and optional elements, as well as mandatory and optional attributes. The top-level of the GenCAM schema contains forty-three elements. The schema notation specifies that the 43 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 writing of all characteristics defined within the requirements stated in this standard.

Each element has a specific function or task respectively.   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 is necessary to produce a given product. The following table lists the children elements of the GenCAM element.
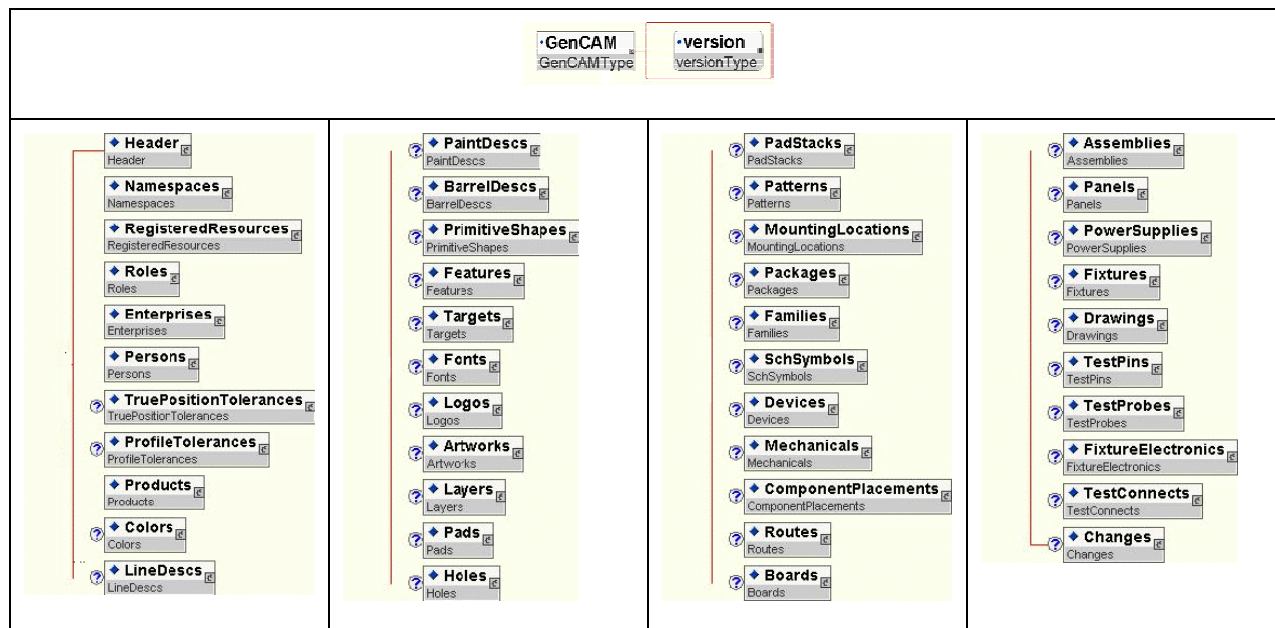
Figure 1 The GenCAM children element

3.1   Rules concerning the use of XML and XML Schema

The rules required to define syntax and semantics of the GenCAM 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 lead to the development of many commercial and open source software tools and libraries that conform to the W3C standards.

A GenCAM file begins with the `<GenCAM revision = "2.0">` tag and end with the `</GenCAM>` tag. The content between these tags must match the xsd definition of the `GenCAM` element as defined by the *GenCAM XML Schema*.

3.1.1   File Readability and Uniformity

A valid GenCAM 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 `</GenCAM>` tag. The checksum is an MD5 message digest algorithm (see *Internet RFC 1321:* http://www.ietf.org/rfc/rfc1321.txt) that is

5

base64 encoded. The checksum starts with the "<" character of the <GenCAM> tag and ending with the ">" character of the closing </GenCAM> tag. The checksum followings immediately after the ">" character of the closing </GenCAM> tag.

The digest provides a 128-bit checksum of the GenCAM file contents. The MD5 signature must be base64 encoded (see *IETF RFC 1421* for the base64 algorithm) to convert the MD5 signature to an 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 a GenCAM file is ".gcx" (*GenCAM XML Schema*).

3.1.4   File Remarks

GenCAM 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 a GenCAM 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 character used in hundreds of written languages. The XML standard allows several of the Unicode encoding formats to be used in an XML file. This revision of IPC-2511 requires the use of the UTF-8 character encoding of the *Unicode* character set.

3.2   Data Organization and Identification Rules

The GenCAM standard uses a namespace mechanism for XML instance files that is similar to the XML namespace mechanism that was created for managing XML meta-data namespaces. The instance file namespace mechanism prevents collisions between the names used by the different products within a single file. This partitioning of namespaces is necessary because the GenCAM file may contain information describing an arbitrary collection of products. (Boards, assemblies, panels, and fixtures are products allowed in a GenCAM file.) For example, a file could contain descriptions for building multiple electronic assemblies that are manufactured on separate panels. This mechanism also prepares the way for a distributed database of GenCAM design data in which the data can be trusted to be universally unambiguous.

3.2.1   Naming elements within a GenCAM file

The concept of GROUP as defined in IPC-2511A was created to allow separation of namespaces in a GenCAM file. This capability was created to allow panel to be created that contain multiple boards. Since two boards may reuse the same identifier, e.g. "U1", "R1", it must be possible to separate names in the file into namespaces. The namespaces capability of the GROUP concept still exists in the XML release of GenCAM, but the implementation has been changed to make namespaces consistent with XML usage. The new implementation borrows the notation used by

XML namespaces. The new system simplifies managing names and makes the new GenCAM format consistent conventional XML usage.

There are two types of names used to name top-level objects (elements instances) in a GenCAM file. The first type of name is a `qualifiedName` type. This type includes a prefix in the name that corresponds to a namespace within the GenCAM file. The prefix and the globally unique identity of the `Namespace` are declared in the `Namespaces` element. The second type of name is a `shortName` type. This type is required to be unique within the GenCAM file. The syntax restrictions on `shortNames` and `qualifiedNames` assure that all names will be unique as top-level names within a GenCAM file.

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 GenCAM 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 GenCAM standard uses these standard types, but the GenCAM standard has been defined without the use of a namespace prefix for element names within a GenCAM file.

Each of the elements in the schema 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 "*element*", and attributes are declared using the element "*element*".

3.2.3   GenCAM Attribute Base Types

The attribute basic types (SimpleTypes) provided by XML Schema are defined by the W3C. They are easy to distinguish from the IPC 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)*.

The following base attribute types are used to define attributes in the GenCAM schema. The `xsd:string` type is constrained to create GenCAM specific base types for special purpose strings, such as qualifiedName and shortName. The rules for special number types and the GenCAM date format are also defined.

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 $3.4 \times 10 - 38 \leq value \leq 3.4 \times 10 + 38$. 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 $0 \leq value \leq 2147483647$ (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 $1 \leq value \leq 2147483647$ (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 GenCAM:
	
	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>
<tr><td colspan="2" align="center">**Basic Types Defined by IPC**</td></tr>
<tr>
<td>qualifiedName</td>
<td>

The `qualifiedName` data type is a data type defined for GenCAM. The type is a restricted `xsd:string` data type where the pattern of the string must match the regular expression "[a-zA-Z][a-zA-Z0-9_-]*:.+".

The definition of the `qualifiedName` data type is:

```
<xsd:simpleType name = "qualifiedName">
  <xsd:restriction base = "xsd:string">
    <xsd:pattern value = "[a-zA-Z][a-zA-Z0-9_-]*:.+"/>
  </xsd:restriction>
</xsd:simpleType>
```

An example of a string that matches the pattern is: "`prefix:name`". The "`prefix`" is a `Namespace` name. The "`name`" is the name of an object within the `Namespace`.

</td>
</tr>
<tr>
<td>point</td>
<td>

The `point` data type is a data type defined for GenCAM. This type is defined as an `xsd:simpleType` that consists of a list of two double separated by whitespace inside of a single string. The first double in the list is the X coordinate. The second double in the list is the Y coordinate. For example:

p1="42.666 3.01e-1" would define X=42.666 and Y=3.01e-1

</td>
</tr>
<tr>
<td>nonNegativeDouble</td>
<td>

The `nonNegativeDouble` data type is defined for GenCAM. The type restricts an xsd:double to positive numbers, inclusive of 0. The non-negative range of values for IEEE doubles is defined as $0.0 \leq value \leq 3. 4x1038$

</td>
</tr>
<tr>
<td>pinName</td>
<td>

The `pinName` data type is a data type defined for GenCAM. The type is a restricted `xsd:string`.

The definition of the pinName data type is:

```
<xsd:simpleType name = "pinName">
    <xsd:restriction base = "xsd:string"/>
</xsd:simpleType>
```

</td>
</tr>
<tr>
<td>xpath</td>
<td>

The `xpath` data type is a data type defined for GenCAM. The type is a restricted `xsd:string` data type where the pattern of the string must be a legal Xpath as defined in W3C http://www.w3.org/TR/xpath.

</td>
</tr>
<tr>
<td>shortName</td>
<td>

The `shortName` data type is a data type defined for GenCAM. The type is a restricted `xsd:string` data type where the pattern of the string must match the regular expression "[a-zA-Z][a-zA-Z0-9_-]*".

The xsd definition of the shortName data type is:

```
<xsd:simpleType name = "shortName">
  <xsd:restriction base = "xsd:string">
    <xsd:pattern value = "[a-zA-Z][a-zA-Z0-9_-]*"/>
  </xsd:restriction>
</xsd:simpleType>
```

An example of a string that matches the pattern is "`bob_24`".

</td>
</tr>
<tr>
<td>`mimeType`</td>
<td>

The `mimeType` data type is a restricted `xsd:string` type that matches IETF MIME type definitions. (e.g. text/html, application/postscript)

</td>
</tr>
</table>

3.2.4   Coordinate System and Transformation Rules

Any geometry defined in a GenCAM 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 .

Coordinates are defined by attributes of type `point`. The definition of a `point` is as follows:

```
<xsd:simpleType name = "point">
    <xsd:list>
        <xsd:simpleType>
            <xsd:restriction base = "xsd:double">
                <xsd:length value = "2"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:list>
</xsd:simpleType>
```

The definition of a Line element can be used to illustrate the use of two points.

An instance of a line would look as follows:

```
<Line startXY="1.0 2.0" endXY="4.0 5.0"/>
```

In this example the line begins at a coordinate of x=1.0 and y=2.0 and the line ends at a coordinate of x=4.0 and y=5.0.

The illustration in Figure 2 provides a perspective drawing of a board and a coordinate system. Each product in a GenCAM 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.
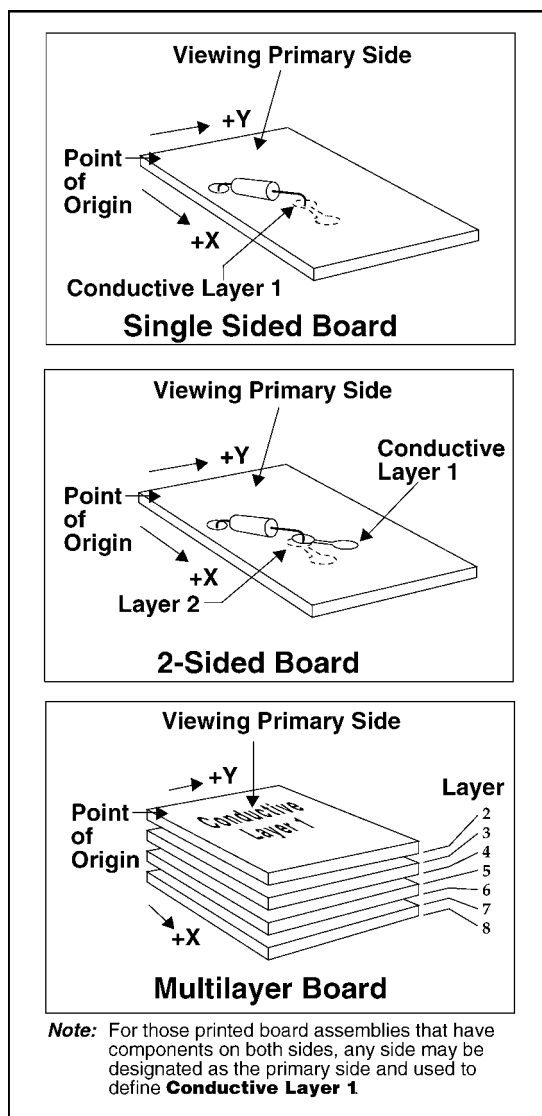
Figure 2 Printed board viewing

3.2.5   Transformation Characteristics

There are four types of transformations used in a GenCAM file. The most general, `Xform`, will be described first. The three additional types of transformations are constrained versions of the `Xform` transform (see table 1). The order of attributes does not define the execution order. The sequence of execution is to mirror, rotate, translate and then scale. The execution order of nested transformations starts at the origin of the outer most product definition and move up the tree of nested geometric elements.

Table 1 Types of Constrained Transformations

Transformation	x	y	rotation	mirror	scale
Xform	O	O	O	O	O
Place	O	O	O	O	
Position	O	O	O		
Offset	O	O			

Note: Mandatory (M), Optional (O)

3.3   Elements used throughout the GenCAM Schema

Several elements are used frequently as children elements. These elements will be defined in this section. When the elements are subsequently used the definitions from this section apply.

3.3.1   The Xform Transformation

The `Xform` element defines a transformation that is used throughout this specification to define the location and orientation of physical features. Primitives in the GenCAM file are manipulated relative to their local Cartesian coordinate system by the values set in the transform. The `Xform` transform defines manipulation of placement, rotation, scaling, and mirror. (see Figure 3)
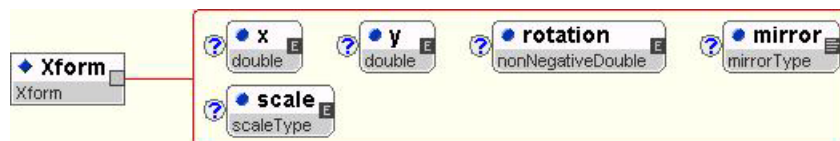
Figure 3 Xform

3.3.1.1   The x and y attribute

The `x` and `y` 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 in GenCAM 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.

3.3.1.2   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 GenCAM file to units of either degrees, or radians. The `Units` element in the `Header` element section specifies the units of measure. If the angle attribute setting for the file is "RADIANS" then the range of the rotation parameter is `0.0   <=   number   <= 6.28318…  (2` . If the angle attribute setting for the file is "DEGREES" then the range of the rotation parameter is `0.0   <=   number <= 360.0`. Positive rotation is always counter-clockwise as viewed from the board TOP (primary

13

side), even if the component being rotated is on the board BOTTOM (secondary side). Rotation defaults to 0.0, and can be applied to text, or any physical shape.

3.3.1.3   The mirror attribute

The `mirror` attribute is of type `mirrorType`. This type is an enumerated `xsd:string` type with a value of either MIRROR or NOMIRROR. The default value is NOMIRROR. When `mirror` is set to MIRROR 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 4. The example shows a unique artwork (14-pin DIP device) placed on the top and bottom of a board at 90-degree rotations.
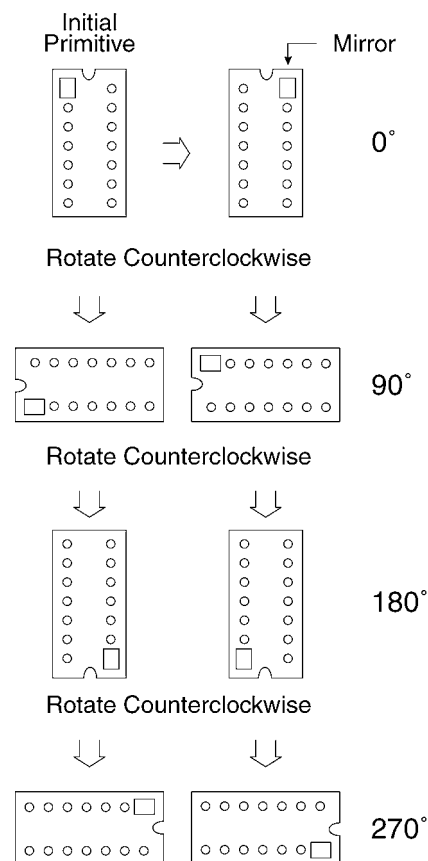
Figure 4 Mirror and Rotation Diagram

3.3.1.4   The scale attribute

The `scale` attribute is of data type `scaleType`. This is a nonNegativeDouble 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.

3.3.2   The *Place* Transformation

The `Place` transformation is identical to `Xform` with the scale attribute fixed at 1.0. The intent of `Place` is to prevent CAD or CAM systems from scaling those items where a change of scale is not appropriate.
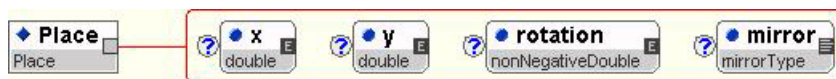
The attributes are defined in 3.3.1.

3.3.3   The *Position* Transformation

The `Position` transformation is identical to `Place` with the `mirror` attribute fixed as NOMIRROR. The intent of `Position` is to prevent CAD or CAM systems from scaling and mirroring those items where a change of scale or orientation is not appropriate.
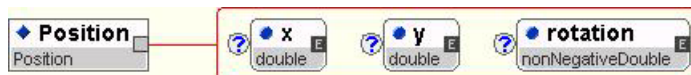
The attributes are defined in 3.3.1.

3.3.4   The *Offset* Transformation

The `Offset` transformation is identical to `Position` with the rotation attribute set to 0.0. The intent of restricting the transformation to just `Offset` is to prevent CAD or CAM systems from scaling, mirroring and rotating those items where a change of scale, rotation, or orientation is not appropriate.
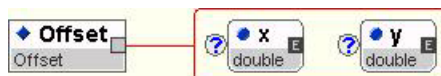
The attributes are defined in 3.3.1.

3.3.5   Text

When text is to be drawn on a product or a drawing the definition includes a bounding rectangle for the text. The lower xy coordinate `p1` and the upper xy coordinate `p2` define the bounding rectangle. All portions of the text, including the line width of the strokes of the text, must fit within the bounding rectangle. Any portion of a character exceeding the perimeter of the bounding rectangle will be clipped at the boundaries of the bounding rectangle.

Attributes	Requirement	Description
textString	REQUIRED	the text phrase (case-sensitive) in accordance with the language element of the Header element
p1	REQUIRED	lower left hand corner of the bounding rectangle.
p2	REQUIRED	upper right hand corner of the bounding rectangle.
fontRef	OPTIONAL	if other than Helvetica is used
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.

The diagram and the description describe the general case for how text is to be draw. There are two variations on the Text type; TextNamedWithLayers and TextWithLayer. These types add additional attributes for some uses of Text. The id is added for Panel and Board because the Text is defined as a top-level element and as such needs to have an id. The layersRef attribute is added in places where the layer on which the Text is drawn is not managed by the containing element, or in the case of Drawing, where there isn't a layer required.

The content of Text/@string should be defined to be enclosed in the textbox as illustrated in Figure 5. 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.
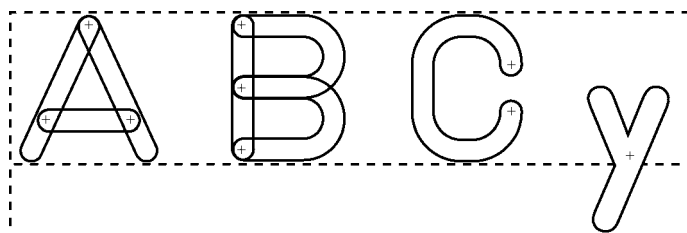
Figure 5 Bounding rectangle to round end character relationships

3.3.5.1   Text transformations

Text character dimensions are constrained by the bounding rectangle as illustrated in Figure 6. Character height is expressed in incremental units of the dimensional characteristics of the file, through the limits (xy coordinates) of the bounding rectangle. Both upper and lower case letters must be inside the bounding rectangle. Included in this requirement are the extensions of such descending letters as lower case "g", "q", "y", "j" and "p". The bounding rectangle of Text is defined relative to the local coordinate system. The Place transform is applied to the bounding rectangle and the text contained within the rectangle. The bounding rectangle is mirrored, then rotated, and then translated per the Place element within the Text element. The text is drawn relative to the bounding rectangle.
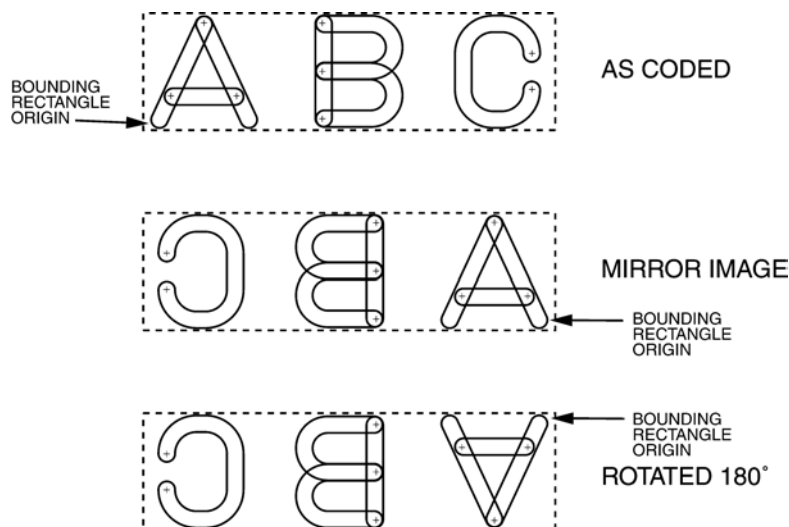
Figure 6 Text transformation examples

The bounding rectangle is rotated 30° about the lower xy coordinate. (see figure 7)
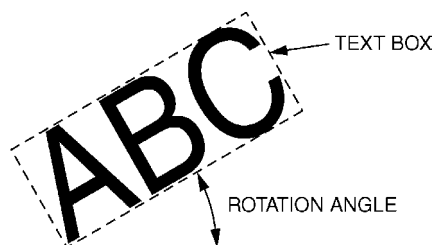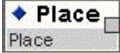
Figure 7 Rotation Angle

4   GenCAM XML Element Definitions

The remainder of the document will define and describe each of the elements that are found in the XML Schema for GenCAM. The order of the definitions is based on the order in which the elements will appear in the file. This order ensures that nothing is referenced prior to being defined.   This also means that the interesting content, the product definitions and test configuration information will appear late in the file. The file order starts with elements that define reusable geometry objects. The elements used to define routes and mounting locations are next. This is followed by elements used to define packages, devices, and component placement. Finally, elements are defined that configure products.

The following table is provided as a convenient reference. It summaries the instructions and definitions for frequently used elements. Consult this table for the definition of any element stub that is not defined for a diagram. Only the stubs of these elements will appear in the element diagrams defined throughout this document. When a stub for the elements, or the substitution elements is present the note in the second column applies. This note references the section of the document where the stub is fully defined. The definitions will not be repeated or referenced in the remainder of this document.

◆ **ClosedShape** ☐ ClosedShape	The elements that can be substituted for the ClosedShape element are defined in the table of *ClosedShape Substitution Elements* defined in Section 4.17
◆ **Offset** ☐ Offset	The transformation defined by the Offset element is to be applied shape element containing the Offset element prior to drawing containing shape. The attributes of the Offset are defined in Section 3.3.4
◆ **Place** ☐ Place	The transformation defined by the Place element is to be applied shape element containing the Place element prior to drawing containing shape. The attributes of the Place are defined in Section 3.3.2
◆ **PolygonBuilder** ☐ PolygonBuilder	The elements that can be substituted for the PolygonBuilder element are defined in the table of *PolygonBuilder Substitution Elements* defined in Section 4.16
◆ **PolylineBuilder** ☐ PolylineBuilder	The elements that can be substituted for the PolylineBuilder element are defined in the table of *PolylineBuilder Substitution Elements* defined in Section 4.18
◆ **Position** ☐ Position	The transformation defined by the Position element is to be applied shape element containing the Position element prior to drawing containing shape. The attributes of the Position are defined in Section 3.3.3
◆ **ShapeBuilder** ☐ ShapeBuilder	The elements that can be substituted for the ShapeBuilder element are defined in the table of *ShapeBuilder Substitution Elements* defined in Section 4.19
◆ **Text** ☐ Text	The definition of the Text element is in Section 3.3.6
◆ **Xform** ☐ Xform	The transformation defined by the Xform element is to be applied shape element containing the Xform element prior to drawing containing shape. The attributes of the Xform are defined in Section 3.3.1

4.1   GenCAM

The top-level element in a GenCAM file is the `GenCAM` element. The `version` attribute of the GenCAM element is shown in Figure 1. This is a string data type that must match the regular expression pattern `'B-2.[0-9]+'`. The version number encodes the version of the standard to which the file data conforms. It is a mandatory attribute used to maintain the relationship between the information represented in the file and the specific release of the standard. GenCAM instance files that conform to the base document for the "B" revision of IPC-2511 must set the version attribute to `"B-2.0"`. When industry requests for extensions or corrections to this base document are created through the release of addendum the new revision of the file format will be indicated by incrementing the least significant digit of the revision number, e.g., `"B-2.1"`, `"B-2.2"`. A future release of this base document would increment the letter and the most significant number in the version attribute, e.g., `"C-3.0"`.

4.2   Header

The `Header` element is mandatory in a GenCAM file. `Header` has four elements that must appear in the order specified in the diagram. The `Language` and `Units` sub-elements are optional and have default values.
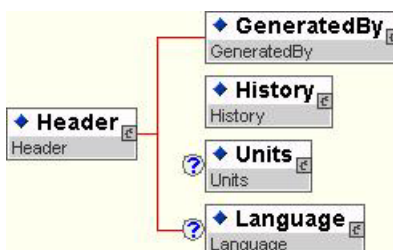
4.2.1   GeneratedBy

The `GeneratedBy` element defines the characteristics of what system generated the file according to the following requirements. The `GeneratedBy` element has two attributes and a sub-element. The attributes and sub-elements of a `GeneratedBy` element are defined as follows:

Attributes	Requirement	Description
`softwarePackage`	REQUIRED	A string describing the software package and platform i.e., "Mentor Board Station"
`revision`	REQUIRED	A string information providing the revision of the software package, i.e., "3.4"
`certicationStatus`	REQUIRED	An enumerated string of any of the fixed field characteristics: ALPHA | BETA | SELFTEST | CERTIFIED

certificationCategory	OPTIONAL	Relates to the type of equipment being certified and the functions that it performs. It can be one or more of the following: GENERALASSEMBLY | ASSEMBLYPANEL | ASSEMBLYPREPTOOLS | GLUEDOT | SOLDERSTENCILPASTE | COMPONENTPLACEMENT | MECHANICALHARDWARE | ASSEMBLYFIXTUREGENERATION | ASSEMBLYTESTGENERATION | ASSEMBLYTESTFIXTUREGENERATION | PHOTOTOOLS | BOARDFABRICATION | BOARDPANEL | BOARDFIXTUREGENERATION | BOARDTESTGENERATION | FABRICATIONDRAWING | ASSEMBLYDRAWING | SCHEMATICDRAWINGS | DETAILEDDRAWING | SPECSOURCECONTROLDRAWING | SINGLEBOARDPARTSLIST | MULTIBOARDPARTSLIST

4.2.2   Units

The Units element defines the unit of measure for length and angle used throughout the GenCAM file. The attributes of a Units element are defined as follows:

Attributes	Requirement	Description
length	DEFAULT	Defines the unit of length for the file as one of: MM|UM|INCH. The default is MM
angle	DEFAULT	Defines the angular unit of measure as one of: DEGREES | RADIANS The range of values for angle measure is defined as follows: RADIANS: $0.0 <= number <= 2\pi$ DEGREES: $0.0 <= number <= 360.0$ The default is DEGREES

4.2.3   History

The `History` element provides a sequential change number for the GenCAM file. The number is changed every time the controlled version of the GenCAM file is modified. Only the file owner is allowed to change the value of `History/@number`. The attributes of a `History` element are defined as follows:
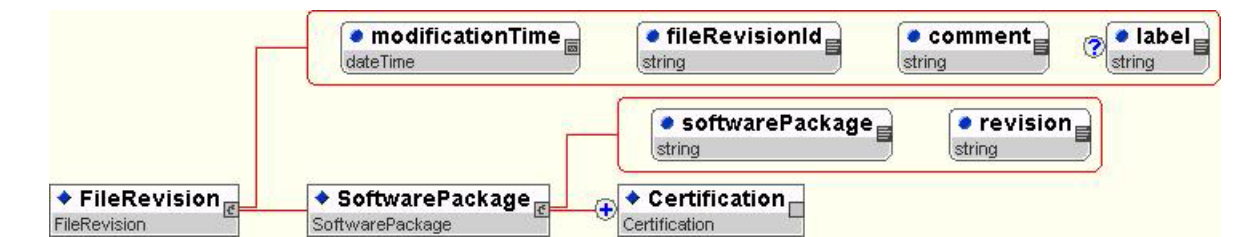
Attributes	Requirement	Description
number	REQUIRED	The revision number of the GenCAM file. The content of this number is defined and controlled by the file owner.
origination	REQUIRED	The timestamp recorded when the GenCAM file was first created.
lastChange	REQUIRED	The timestamp recorded when the History number was last incremented.
externalConfigurationControl EntryPoint	OPTIONAL	A URI referencing a configuration control system that "owns" the GenCAM file contents.

4.2.4   FileRevision

The FileRevision element tracks changes to the GenCAM 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.

Attributes	Requirement	Description
modificationTime	REQUIRED	The timestamp of when the software safed the revision of the file.
fileRevisionId	REQUIRED	An identifier for the revision. This value may be supplied by a revision control system such as RCS, CVS, or SCCS.
comment	REQUIRED	A short description of the revision, such as a changes statement entered by RCS or SCCS.

label	OPTIONAL	A label that can be applied to a branch head. The label can be used to associate a file revision of special
softwarePackage	REQUIRED	The software package that wrote this revision of the file.
revision	REQUIRED	The revision of the software that wrote the file.

4.2.5   Language

The `Language` element declares the written language used in the GenCAM file. The only parameter of a `Language` element is defined as follows:

Attributes	Requirement	Description

`lang`	REQUIRED	Internet Engineering Task Force (IETF) defines language codes in http://www.ietf.org/rfc/rfc1766.txt. (See the IETF for the full specification of the requirements.) The basic format is a language tag with an optional country code suffix. The syntax defined by RFC-1766 is:
		
		`The syntax of this tag in RFC-822 EBNF is:`
		
		`    Language-Tag = Primary-tag *( "-" Subtag )`
		`    Primary-tag = 1*8ALPHA`
		`    Subtag = 1*8ALPHA`
		
		`This means the value can be a string of up to 8 ASCII alphabetic characters followed by zero or more optional "-" characters that are followed by a string of up to 8 alphabetic characters`
		
		`Some Examples:`
		
		`<Language lang="zh" />  - uses Chinese`
		`<Language lang="en" />  - uses English`
		`<Language lang="fi" />  - uses Finnish`
		`<Language lang="fr" />  - uses French`
		`<Language lang="ge" />  - uses German`
		`<Language lang="ja" />  - uses Japanese`
		`<Language lang="ko" />  - uses Korean`
		`<Language lang="pt" />  - uses Portuguese`
		`<Language lang="ru" />  - uses Russian`
		`<Language lang="es" />  - uses Spanish`
		`<Language lang="sv" />  - uses Swedish`
		
		`<Language lang="no-bokmaal" />`
		`    - one of two offcial versions of Norwegian`
		
		The first part half of the tag is the language code as defined in ISO-639. The code values are accessible from http://www.oasis-open.org/cover/iso639a.html. The interpretation of the codes are case insensitive, but lowercase is the prefered notation according to the IETF. The remainder of the code define the country and region of dialect. The country codes are defined in ISO-3166.

4.3   Namespaces

`Namespaces` are used to eliminate the naming conflicts that are introduced by the reuse of names in the definition of products. (For example, a reference designator of "U1" and "R1" is likely to appear in many products.) The `Namespaces` mechanism can be used to guarantee that names used in the declaration of product will be universally unique. The `Namespaces` element defines the list of `Namespace` elements that will be used in the GenCAM file.
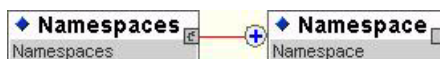
4.3.1   Namespace

A `Namespace` associates a short prefix with a long, globally unique identifier. It is possible to eliminate conflicts caused when two products defined in a GenCAM file share a common base name by using a separate prefix for the name used in each product in the GenCAM file. The globally unique identifier extends this conflict resolution to ensure the names are also globally unique.

Attributes	Requirement	Description

prefix	OPTIONAL	All `prefix` strings (the character sequence before the ':' in a qualifiedName) are declared as `Namespace` prefixes. For example, in the definition of a `Device` the `id` and `packageRef` attributes use qualifiedName types.
		
		 `<Device id="dev1:LM555"`
		
		 `packageRef="pk:8pinDIP"…>`
		
		 The "dev1" string in id and the "pk" string in `packageRef` are prefix strings. All such prefix strings used in a GenCAM file must be declared as a `Namespace`.

uri	REQUIRED	The URI of a GenCAM namespace is an abbreviation of the namespace URI, just as XML names-space prefixes are abbreviations of XML name-space uri. The full name of the GenCAM object is a concatenation of the URI with the string following the ':' in the `qualifiedName`.
description	REQUIRED	A description of the source of the GenCAM name-space can be added to assist in understanding the source of the names used in the GenCAM file.

4.4   RegisteredResources

The RegisteredResources element defines a list of all RegisteredResource elements used within the GenCAM file.
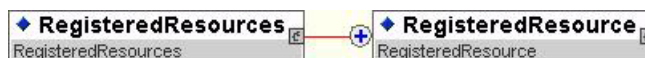
4.4.1   RegisteredResource

Attributes	Requirement	Description
id	REQUIRED	The id attribute is referenced from elements within the GenCAM file. A referencing element points to the RegisteredResource id to define the allowed values and data types for an associated attribute in the referencing element. The RegisteredResource defines attribute characteristics that are within the control of other IPC committees. The definition pointed to by the url attribute may include a list of the allowed enumerated value or minimum and maximum range for the values of the RegisteredResource.
url	REQUIRED	The url attribute contains a URL for the definition of the RegisteredResource.
description	OPTIONAL	A description of the local interpretation of a RegisteredResource can be added to the definition.

4.5   Roles

The Roles element defines a list of all Role elements used within the GenCAM file.

25
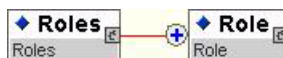
4.5.1   Role

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.

Attributes	Requirement	Description
id	REQUIRED	The `id` uniquely identifies a role type used by the enterprise. The `id` is a `shortName` data type (a restricted `xsd:string`) that must be unique within the global (top-level) namespace of the GenCAM file. The standard IPC role types are defined as follows:
		SENDER – Identifies the person sending out the GenCAM file.
		OWNER – Identifies the person who maintains the configuration management of the GenCAM file and has the right to increment the file history number of the GenCAM file.
		RECEIVER – Identifies the person receiving the GenCAM file.
		DESIGNER – Identifies the designer of the product described in the GenCAM file.
		ENGINEER – Identifies the engineer who is responsible for the product described in the GenCAM 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.
description	OPTIONAL	The `description` attribute defines a role within an enterprise. (The description is optional if the IPC definition is to be used.)
publicKey	OPTIONAL	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.
authority	OPTIONAL	The access level associated with this role as defined by the system referenced by `externalConfigurationControlEntryPoint`

4.6   Enterprises

The `Enterprises` element defines a list of all `Enterprise` elements used within the GenCAM file.
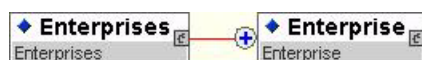
4.6.1   Enterprise

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

Attributes	Requirement	Description
`id`	REQUIRED	The `id` uniquely identifies an enterprise throughout the GenCAM file. The `id` is a `shortName` data type (a restricted `xsd:string`) that must be unique within the global (top-level) namespace of the GenCAM file. (Suggest "XYZ", "ACME"...)
`name`	REQUIRED	The full name of the enterprise.
`address1`	OPTIONAL	The street address of the enterprise.
`address2`	OPTIONAL	Additional address information for the enterprise.
`city`	OPTIONAL	The city.
`stateProvince`	OPTIONAL	The state or province.
`country`	DEFAULT	The two-letter ISO country code from the ISO 3166 standard. (See ftp://info.ripe.net/iso3166-countrycodes). The default country is "us"
`postalCode`	OPTIONAL	The postal code.
`phone`	OPTIONAL	The phone number.
`fax`	OPTIONAL	The phone number of the fax machine.
`email`	OPTIONAL	The email address.
`url`	OPTIONAL	The Internet HTTP Web address of the enterprise.
`code`	REQUIRED	The value of the specified `codeType` for the enterprise.

codeType	DEFAULT	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)
		If no CAGE or DUNS code is available use "—NONE--" as the value of the code attribute.
roleRef	REQUIRED	A role within the organization.

4.7   Persons

The `Persons` element defines a list of all `Person` elements used within the GenCAM file.

4.7.1   Person

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

Attributes	Requirement	Description
id	REQUIRED	A string that uniquely identifies the person throughout the GenCAM file. The `id` `shortName` must be unique within the global (top-level) namespace of the GenCAM file. (Suggest "sally", "bob"…)
name	REQUIRED	The person's full name.
enterpriseRef	REQUIRED	The `shortName` of the person's company or enterprise; if no enterprise exists, the term "SELF" should be used.
title	OPTIONAL	The job title of the person.
email	OPTIONAL	The email address of the person.
phone	OPTIONAL	The phone number of the person.
fax	OPTIONAL	The fax machine phone number of the person.

mailstop	OPTIONAL	The World Wide Web URL of the person.
publicKey	OPTIONAL	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.
roleRef	REQUIRED	The role of the person.

4.8   TruePositionTolerances

The TruePositionTolerances element defines a list of all TruePositionTolerance elements used within the GenCAM file.
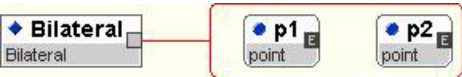
4.8.1   TruePositionTolerance

The TruePositionTolerance element defines the manufacturing tolerance range for positioning geometry in a product definition. An indepth explaination of product tolerancing is available in the IPC-2221 standard. The concepts are illustrated in Figure 8. The attributes of a TruePositionTolerance element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the TruePositionTolerance within the GenCAM file. The id is referenced by locationTolRef attributes in order to associate the tolerance definition with geometries to which they apply.
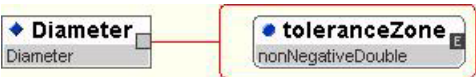
4.8.2   Bilateral

The Bilateral element defines a square tolerance zone as the location for the centroid of any feature. This is usually expressed as a plus or minus tolerance applied to the true position location. The following describes the upper right (posTol) and lower left (negTol) corner of the tolerance zone taken from the true position of the feature.

p1	REQUIRED	The lower left corner of the tolerance zone
p2	REQUIRED	The upper right corner of the tolerance zone

4.8.3   Diameter

The Diameter element defines the tolerance zone as a diameter from the true position coordinate of the feature.

toleranceZone	REQUIRED	a dimension that represents the diameter of true position (DTP) tolerance zone

4.8.4   MaximumMaterialConditionDiameter

The MaximumMaterialConditionDiameter element defines the diameter of true position with the allowance that if the feature positioned at that location is at a Maximum Material condition the additional difference between Maximum and Least material condition can be added to the tolerance zone.

mmcToleranceZone	REQUIRED	diameter of true position increased by the difference between MMC and LMC

Bilateral Tolerance Zone

The shaded square represents the tolerance
zone of a hole with a positioned tolerance of 0.13 ± .

Positional Tolerance Zone

By using the positional tolerance shown in Detail A, a 0.18
diameter tolerance zone is established. The tolerance zone
is increased 57%.

Bonus Tolerance Based on Maximum Material Concept

By modifying the positional tolerance to apply at maximum
material condition, as shown in detail B, the tolerance
zone increases as the measured hole size deviates
from its minimum size (maximum material condition).
In this example, the tolerance zone can increase to 0.25.

Figure 8 Advantages of positional tolerance over bilateral tolerance, mm

4.9   ProfileTolerances

The ProfileTolerances element defines a list of all ProfileTolerance elements used within the
GenCAM file.
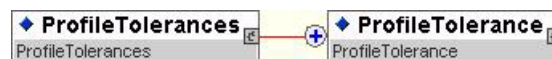
4.9.1   ProfileTolerance

The `ProfileTolerance` element defines the rectangular manufacturing tolerance range for positioning geometry in a product definition. An indepth explaination of product tolerancing is available in the IPC-2615 standard. The attributes of a `ProfileTolerance` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `ProfileTolerance` within the GenCAM file. The `id` is referenced by `profileTolRef` attributes in order to associate the tolerance definition with geometries to which they apply.
finishedLMC	REQUIRED	The permitted variation from nominal to the least material condition (LMC) of a feature of the product.
finishedMMC	REQUIRED	The permitted variation from nominal to the maximum material condition (MMC) of a feature of the product.
startLMC	OPTIONAL	The permitted variation from nominal to the least material condition (LMC) of a feature or the product prior to plating, coating or additional material removal.
startMMC	OPTIONAL	The permitted variation from nominal to the maximum material condition (MMC) of a feature or the product prior to plating, coating or additional material removal.

4.10   Products

The `Products` element defines a list of all `Product` elements used within the GenCAM file.
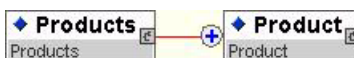
4.10.1   Product

The `Product` element identifies a product that is described in the GenCAM file. The attributes of a `Product` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `shortName` that uniquely identifies the Product within the GenCAM file. The `id` is referenced by attributes of with types `boardRefType`, `assemblyRefType`, `panelRefType`, and `fixtureRefType` in order to associate the product administrative definition with product configuration definition.
name	REQUIRED	A descriptive name or title for the product.
number	REQUIRED	The part number of the panel or subpanel.
revision	OPTIONAL	The revision level of the panel or subpanel.
type	REQUIRED	The type of the product. One of BOARD | PANEL | ASSEMBLY | FIXTURE.

type	REQUIRED	The type of transaction. A transaction may be a purchase order (PO), a request for quote (RFQ), a request for proposal or information (RFP), a request for audit (RFA), or a change order (CO).
number	REQUIRED	An externally generated identifier for the transaction. GenCAM does not interpret this identifier. The value of this attribute can be used to link the GenCAM transaction with an external system, such as a factory ERP or MES information system.

date	REQUIRED	The date on which the transaction was initiated.

quantity	REQUIRED	The quantity of the item to be delivered on the `deliverableDate` **date.**
deliverableDate	REQUIRED	The date by which the item is to be received.

enterpriseRef	REQUIRED	The `enterpriseRef` attribute references the `Enterprise`/@id to define the `preferred` vendor(s) for the product.
personRef	OPTIONAL	The `personRef` attribute references the `Person`/@id to identify the contact point at the preferred vendor.

4.11   Colors

The Colors element defines a list of all Color elements used within the GenCAM file.

4.11.1   Color

The Color element defines a color that is referenced throughout the GenCAM file. The 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:

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the Color within the GenCAM file. The id is referenced by colorRef attributes in order to associate the color definition with geometries to which it applies.
r	REQUIRED	Defines the red color intensity as a value between 0 and 255.
g	REQUIRED	Defines the green color intensity as a value between 0 and 255.
b	REQUIRED	Defines the blue color intensity as a value between 0 and 255.

Examples of Colors

<Colors>
 <Color id = "IPCstd:white" r = "255" g = "255" b = "255"/>
 <Color id = "IPCstd:black" r = "0" g = "0" b = "0"/>
 <Color id = "IPCstd:red" r = "255" g = "0" b = "0"/>
 <Color id = "IPCstd:green" r = "0" g = "255" b = "0"/>
 <Color id = "IPCstd:blue" r = "0" g = "0" b = "255"/>

4.12   LineDescs
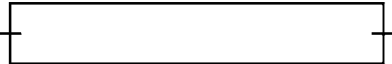
The `LineDescs` element defines a list of all `LineDesc` elements used within the GenCAM file.
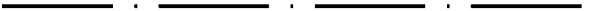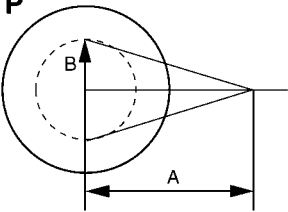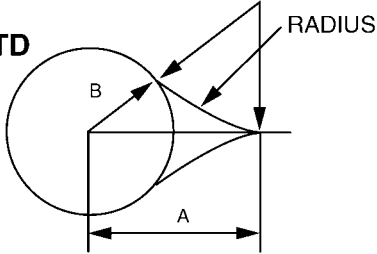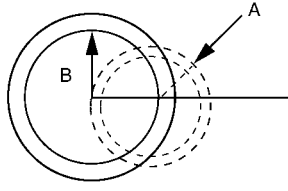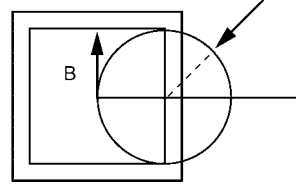
4.12.1   LineDesc

The `LineDesc` element defines the characteristics of a line. When a `LineDesc/@id` is referenced by an enclosing element contains a `lineDescRef` attribute the characteristics of the `LineDesc` are applied to all the enclosed line segments. The line characteristics includes the line width, the line type, the end characteristics, end modifications, and color. The rules of precedence for `LineDesc` are defined in Figure 10. The attributes of a `LineDesc` element are defined as follows:

Attributes	Requirement	Description
`id`	REQUIRED	The `id` attribute is a qualifiedName that uniquely identifies the LineDesc within the GenCAM file. The `id` is referenced by `lineDescRef` attributes in order to associate the `LineDesc` definition with geometries to which it applies.
`width`	REQUIRED	The line width in length dimension units.
`lineEnd`	DEFAULT	The line end is one of SQUARE, ROUND or NONE. The default is ROUND.
`colorRef`	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
`type`	DEFAULT	The line type is one of SOLID, DOTTED, DASHED, CENTER, PHANTOM or (solid) ERASE. The default is SOLID.
`space`	OPTIONAL	The spacing between dots and dashes. The meaning by `LineDesc/@type` is: DOTTED – the spacing between the dot centers. DASHED – the spacing between the end of one dash and the start of the next dash. CENTER – the spacing between the center of the dot and the line ends of the line segments. PHANTOM – the spacing between the center of the dots and the line ends of the line segments.

36

length	OPTIONAL	The length of dashes. The meaning by `LineDesc/@type` is:
		DASHED – the length of the dashes.
		CENTER – the length of the line segments between the dots.
		PHANTOM – the length of the line segments between the dots.
lineMod	DEFAULT	The line end modification is one of tear-drop (TD), tapered (TP), or one of two sub-land patterns (SL1, SL2). See the following table. The default is NONE.
modEnd	DEFAULT	The ends of the line that are to have `LineDesc/@lineMod` applied. The value is one of START| FINISH | BOTH. The default is BOTH.
		BOTH – modify both start and finish of line ends
		START – modify the start point of a line
		FINISH – modify the last point of a line
dimA	OPTIONAL	The meaning is dependent on the `LineDesc/@lineMod` attribute. The meaning by `LineDesc/@lineMod` is:
		TP – length of the taper fillet as measured from the center of the pad along the x-axis.
		TD – length of the tear-drop as measured from the center of the pad along the x-axis.
		SL1 – radius of the sub-land circle.
		SL2 – radius of the sub-land circle.
dimB	OPTIONAL	The meaning is dependent on the `LineDesc/@lineMod` attribute. The meaning by `LineDesc/@lineMod` is:
		TP – z-axis dimension of the taper fillet as measured at the center of the pad.
		TD – the radius of the pad.
		SL1 – distance measured along the x-axis from the center of the pad to the center of the sub-land.
		SL2 – distance measured along the x-axis from the center of the pad to the center of the sub-land.
dimC	OPTIONAL	The LineDesc/@lineMod attribute. The meaning by LineDesc/@lineMod is:
		TD – radius joining curvature of land to the x-axis.
This diagram illustrates the `LineDesc/@lineEnd` attribute		ROUND
		SQUARE
		NONE

This diagram illustrates the `LineDesc/@type` attribute. Line type attributes, other than SOLID and ERASE, also require attributes to define spacing characteristics for the line segments, dashes and dots of the lines.	SOLID DOTTED DASHED CENTER PHANTOM ERASE
This diagram illustrates the `LineDesc/@lineMod,` `LineDesc/@dimA,` `LineDesc/@dimB,` and `LineDesc/@dimC` attributes taper (fillet) TP,.... tear-drop TD,.... Sub-lands SL1,.... Sub-lands SL2,....	TP TD RADIUS SL1 SL2

The following table defines the rules of interaction between `lineMode` types and the additional line dimensioning attributes.

Table 2 Line Modification Rules

lineMod	dimA	dimB	DimC
TP	required	required	Prohibited
TD	required	required	Required
SL1	required	required	Prohibited
SL2	required	required	Prohibited
--NONE--	prohibited	prohibited	Prohibited

LineDesc Examples:

```
<LineDesc  id = "IPCstd:couryardl" width = "1.0" type = "DOTTED"
           length = "2.5"/>
```

The lines to which it is applied will be a black dotted line, 1.0 mm diameter for the dot, 2.5 mm for the space between dot centers.

```
LINEDESC: "line3", 1.0, , ,CENTER ,1.5, 2.8;
```

The lines to which it is applied will be a repeating black dash, dot, dash line, 1.0 mm wide (this is also the diameter of the dots), 1.5 mm space between dot center and start or ending of dash, and 2.8 mm for length of dash.

```
LINEDESC: "Line9", 1.0, SQUARE , , CENTER, 8.0, 40.0,
              TP, BOTH, 2.5, 1.3;
```

The lines to which it is applied will be a CENTER line, black by default, 1.0 wide, where the length of the dash is 40.0 and the clearance between the center of the dot and the beginning and end of the dash is 8.0. Since the line ends are SQUARE, the dot is also square - 1.0 x 1.0). Both ends of the line segment are tapered.

```
<LineDesc  id = "IPCstd:phantom" width = "1.0" type = "PHANTOM"
           colorRef = "IPCstd:red" lineEnd = "ROUND" space = "8.0"
           length = "25.0" modEnd = "START" lineMod = "TD" dimA = "2.0"
           dimB = "1.7" dimC = ".5"/>
```

The lines to which it is applied will be a red PHANTOM line, 1.0 wide, where the dashes are 8.0 long, and the distance from beginning and end of the dash to the center of the dots, or the distance between dot centers is 8.0. The starting end of the line is a teardrop, while the opposite end is round by default.

4.13   PaintDescs

The `PaintDesc` element defines the fill characteristics that can be applied to a closed shape. The paint description definitions are referenced throughout the GenCAM file using a `paintDescRef` attribute. GenCAM supports the paint types of HOLLOW, FILL, MESH, HATCH and VOID (see Figure 9). The following table defines all the attribute options for `PaintDesc`.

4.13.1   PaintDesc

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a qualifiedName that uniquely identifies the `PaintDesc` within the GenCAM file. The `id` is referenced by `PaintDescRef` attributes in order to associate the `PaintDesc` definition with geometries to which it applies.
type	DEFAULT	The texture of the fill is one of HOLLOW, HATCH, MESH, FILL or VOID. The default is HOLLOW.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
lineWidth	OPTIONAL	The width of the lines used in a HATCH or MESH. (This attribute only applies to the paint types HATCH and MESH.)
pitch1	OPTIONAL	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 paint types HATCH and MESH.)
pitch2	OPTIONAL	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 paint type MESH.)
angle1	OPTIONAL	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 90 degrees. The default is 45 degrees. (This attribute only applies to the paint types HATCH and MESH.)
angle2	OPTIONAL	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 paint type MESH.)
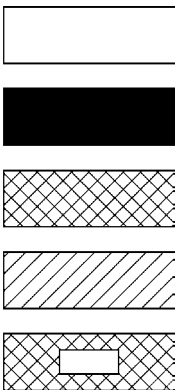
This diagram illustrates the PaintDesc/@type attribute. The LineDesc/@type is set to SOLID in the examples to provide an outline around the shapes that are painted with MESH, HATCH, or VOID.	HOLLOW FILL MESH HATCH VOID
This diagram illustrates the PaintDesc/@type attribute with several different shapes. VOID removes the immediately previous, positive instance. A VOID overlapping a VOID only extends the VOID and does not remove the second positive instance.	circle – FILL rectangle – MESH rectangle – HATCH polygon with VOID

Figure 9 PaintDesc fill definitions

Example:

```
<PaintDesc id="IPCstdFills:compfill" type="HATCH" colorRef="IPCcolors:BLACK"
           lineWidth="0.20" pitch1="0.20" angle1="45" />
```

The closed shape to which it is applied will be a hatched black fill with a line width of 0.20 mm and a pitch between lines of 0.20 mm and an angle of 45 degrees.

4.13.2   Rules of Precedence

Both the `LineDesc` and `PaintDesc` elements have an optional `colorRef` attribute. The addition of an optional `colorRef` attribute, directly on the element instance has the possibility of causing a conflict with the `colorRef` on a definition element. Should this happen, the `colorRef` specified for the more local enhancement element (`LineDesc`, `PaintDesc`) overrides a previously defined value or the default value. The `colorRef` example in figure 10 shows the rules of precedence and indicates how usage textures override referenced textures included in the definition of an artwork or pattern. The `colorRef` attribute is used as an example.
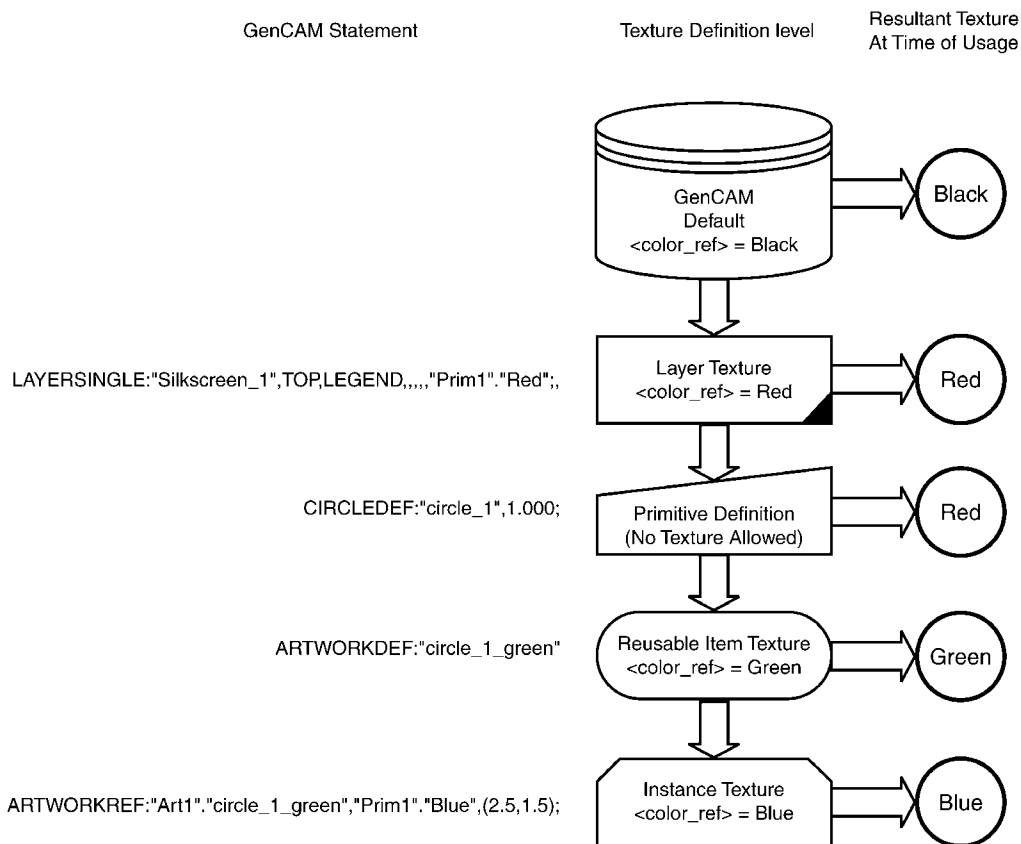
Figure 10 Rules of precedence

4.14   PrimitiveShapes

The physical characteristics of an electronic interconnection product allow the product geometry definition to be represented using a small set of primitive 2-D shapes. It would be possible to represent everything using just polygon and polyline primitives, however, a more concise and descriptive product representation is possible if the base set of primitives also includes the most commonly used closed shapes, such as rectangles, thermals, and circles. The description of the product geometry can more accurately and concisely reflect the product design if standard sized graphical elements, such as pads in a land pattern, can also be defined once and referenced when used. The graphical elements that are defined in `PrimitiveShapes` element enable this reuse. Figure 11 shows the list of primitive shapes that can be defined. These definitions are referenced to create instances of the shapes.
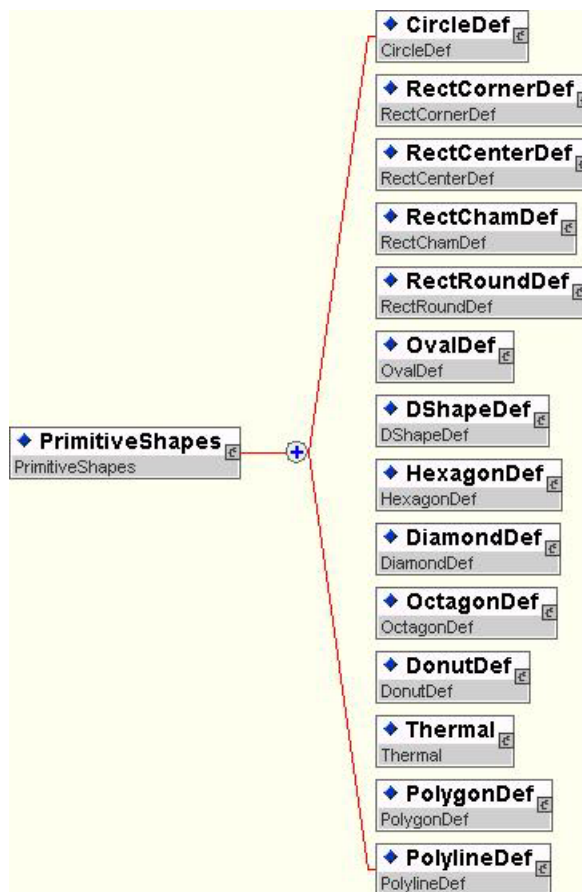
Figure 11 Allowed PrimitiveShape

The graphical elements in `PrimitiveShapes` are defined independent of a local coordinate system. When a primitive shape definition is referenced the location of the instance of the shape is determined by a placement transformation associated with the reference. The placement is always relative to the local coordinate system of the element that references the definition.

4.14.1   CircleDef

A `CircleDef` is a primitive shape that defines a circle by the diameter of the circle. The point of origin is the center of a circle. The circle is illustrated in Figure 12.

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `CircleDef` within the GenCAM file. The `id` is referenced by `circleRef` attributes of a CircleRef element in order to create an instance of the circle.
diameter	REQUIRED	The diameter of the circle.
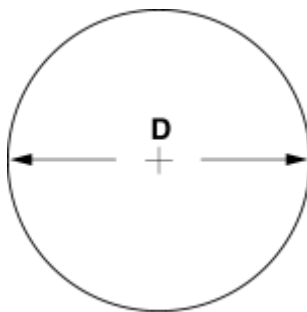
Figure 12 Circle

Examples:

        <CircleDef id="bd1:circ2" diameter="2.0"/>
        <CircleDef id="bd2:hole2" diameter="0.01"/>

4.14.2   RectCornerDef

A `RectCornerDef` is a primitive shape that defines a rectangle by the lower left and upper right corners of the rectangle. The rectangle in Figure 13 illustrates the relationship between the corner coordinates and the rectangle. The point of origin of a `RectCornerDef` rectangle is (0, 0). This is can be coincident with attribute `p1`, the lower left corner of the rectangle, but there is no requirement for `p1` 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 `p1` or `p2`.

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the RectCornerDef within the GenCAM file. The id is referenced by rectCornerRef attributes in order to create an instance of the rectangle.
p1	REQUIRED	The point that defines the lower left corner of the rectangle.
p2	REQUIRED	The point that defines the upper right corner of the rectangle.

Figure 13 A RectCorner Rectangle

Examples:

        <RectCornerDef id = "bd1:rect6" p1 = "1240 3370" p2 = "4535 2355" />

4.14.3   RectCenterDef

A `RectCenterDef` 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 `RectCenterDef` 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 rectangle in Figure 14 illustrates the relationship between the origin and the dimensions of the rectangle.

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the RectCenterDef within the GenCAM file. The id is referenced by rectCenterRef attributes in order to create an instance of the rectangle.
width	REQUIRED	The length of the rectangle about the x-axis.
height	REQUIRED	The length of the rectangle about the y-axis.
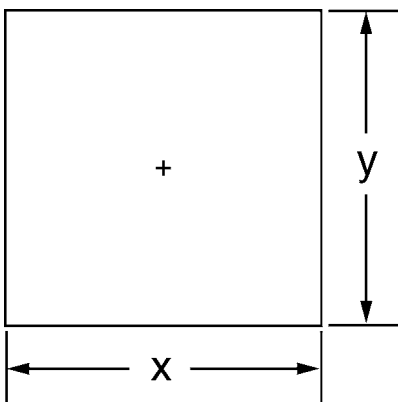
Figure 14 A RectCenterDef Rectangle

Examples:

<u>Examples:</u>

<rectCenterDef id = "bd1:rect3" width = "1.6" height = "2.8"/>

4.14.4   RectChamDef

A RectChamDef is a primitive 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 RectChamDef 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 rectangle in Figure 15 illustrates the relationship between the origin and the dimensions of the rectangle.

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `RectChamDef` within the GenCAM file. The `id` is referenced by `rectChamRef` attributes in order to create an instance of the rectangle.
width	REQUIRED	The length of the rectangle about the x-axis.
height	REQUIRED	The length of the rectangle about the y-axis.
chamfer	REQUIRED	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.

Figure 15 Chamfered Rectangular Primitive

4.14.5 RectRoundDef

A `RectRoundDef` is a primitive 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 `RectRoundDef` 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 rectangle in Figure 16 illustrates the relationship between the origin and the dimensions of the rectangle.

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the RectRoundDef within the GenCAM file. The id is referenced by rectRoundRef attributes in order to create an instance of the rectangle.
width	REQUIRED	The length of the rectangle about the x-axis.
height	REQUIRED	The length of the rectangle about the y-axis.
radius	REQUIRED	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.
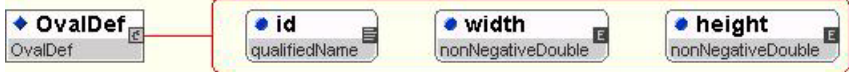
Figure 16 Rounded Rectangular Primitive

4.14.6   OvalDef

An OvalDef is a primitive 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 OvalDef 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 OvalDef is defined in 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. The oval shaped rectangle in Figure 17 illustrates the relationship between the origin and the dimensions of the oval shaped rectangle.

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the OvalDef within the GenCAM file. The id is referenced by ovalRef attributes in order to create an instance of the oval ended rectangle.
width	REQUIRED	The length of the rectangle about the x-axis.
height	REQUIRED	The length of the rectangle about the y-axis. It is an error to define a height greater than the width.

Figure 17 Oval Rectangular Primitive

4.14.7   DShapeDef

A DShapeDef is a primitive shape that defines a rectangle with one of three D-shapes replacing the right side of the rectangle. (The rotation attribute of a transform can be used to position the shaped end of the rectangle at other orientations.) 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 DShapeDef 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 DShapeDef defines special corner shapes for the two corners on the right side. The endShape attribute indicates the type of corner. The D-shaped rectangle in Figure 18 illustrates the endShapes and the relationship between the origin and the dimensions of the D-shaped rectangle.

49

| Attributes	Requirement	Description
id	REQUIRED	The *id* attribute is a `qualifiedName` that uniquely identifies the DShapeDef within the GenCAM file. The *id* is referenced by dShapeRef attributes in order to create an instance of the D-shaped rectangle.
endShape	REQUIRED	One of ROUND | FILLET | CHAMFER. Defines the type of modification that will be made to the corners on the right of the rectangle. ROUND — trim back the right sideof the rectangle to be a semicircle with a radius equal to ½ the width of the rectangle. FILLET — a radius is cut on the top corners of the rectangle. CHAMFER — the top two corners are clipped at a 45 degree angle.
width	REQUIRED	The length of the rectangle about the x-axis.
height	REQUIRED	The length of the rectangle about the y-axis.
corner	OPTIONAL	For FILLET and CHAMFER D-shapes the meaning of the endShape is: FILLET — the radius to be trimmed from the right side corners ofthe rectangle. If the radius is greater than ½ the height or ½ the width then the definition is undefined. CHAMFER — the length along the width and height of each corner between which the rectangle is to be clipped. If the value of chamfer is greater than ½ the height or ½ the width then the definition is undefined. The corners are on the right side.
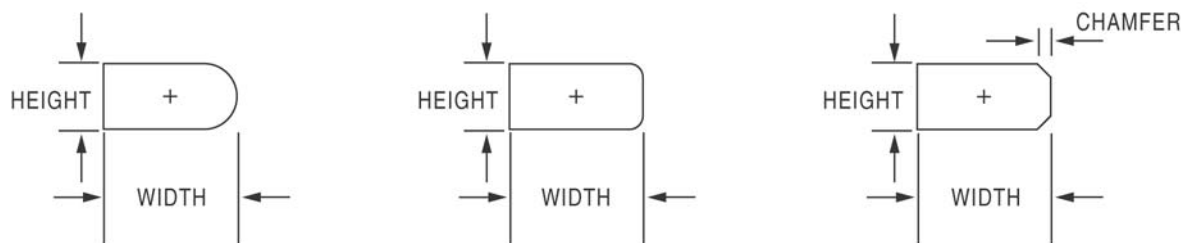
Figure 18 D-Shaped Rectangle Primitives – Round, Fillet, Chamfer

4.14.8   HexagonDef

A `HexagonDef` is a six-sided primitive shape with each of the sides being equal in length and with all angles between adjacent sides also being equal. The only dimension specified is the distance between points on the opposite side of the hexagon. The hexagon is defined with two of the points positioned on the x-axis with the origin at the center of the hexagon. The hexagon is illustrated in Figure 19.

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `HexagonDef` within the GenCAM file. The `id` is referenced by `hexagonRef` attributes in order to create an instance of the hexagon shape.
pointToPoint	REQUIRED	The distance between the two corner points of the hexagon on the x-axis.
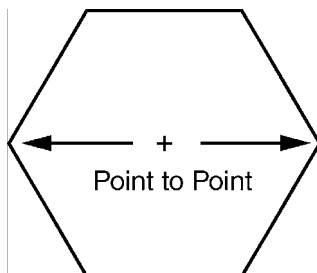
Figure 19 Hexagon Primitive

4.14.9   DiamondDef

A `DiamondDef` is a 4-sided primitive 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 heights 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. An example of the diamond is illustrated in Figure 20.

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `DiamondDef` within the GenCAM file. The `id` is referenced by `diamondRef` attributes in order to create an instance of the `diamond` shape.
width	REQUIRED	The length of the diamond along, and centered on, the x-axis.
height	REQUIRED	The length of the diamond along, and centered on, the y-axis.
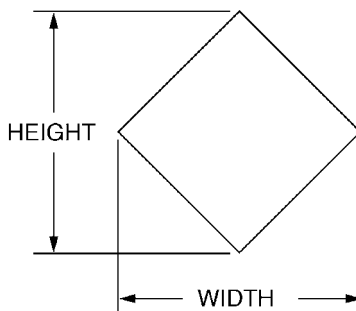
Figure 20 DiamondDef Element

4.14.10 OctagonDef

A `OctagonDef` is a eight-sided primitive shape with each of the sides being equal in length and with all angles between adjacent sides also being equal. The only dimension specified is the distance between points on the opposite side of the octagon. The octagon is defined with two of the points positioned on the x-axis with the origin at the center of the octagon. The octagon is illustrated in Figure 21.

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `OctagonDef` within the GenCAM file. The `id` is referenced by `octagonRef` attributes in order to create an instance of the octagon shape.
pointToPoint	REQUIRED	The distance between the two corner points of the octagon on the x-axis.
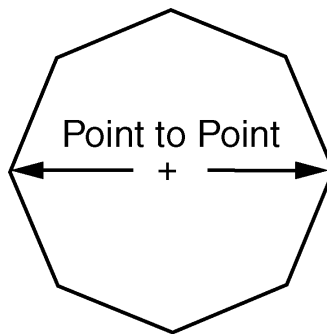
Figure 21 OctagonDef Element

4.14.11 DonutDef

A `Donut` is composed of two identical concentric shapes that are square, round, hexagonal, or octagonal. The center of a `Donut` is also the point of origin of the primitive. Examples of `Donut` shapes are illustrated in Figures 22-25.

Attributes	Requirement	Description
`id`	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `DonutDef` within the GenCAM file. The `id` is referenced by `donutRef` attributes in order to create an instance of the donut shape.
`shape`	REQUIRED	The inner and outer shapes are one of ROUND, SQUARE , HEXAGON or OCTAGON: 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. HEXAGON – The inner and outer shapes are like HexagonDef. OCTAGON – The inner and outer shapes are like OctagonDef.
`outerDiameter`	REQUIRED	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.

innerDiameter	REQUIRED	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.

Figure 22 Round DONUT Primitive
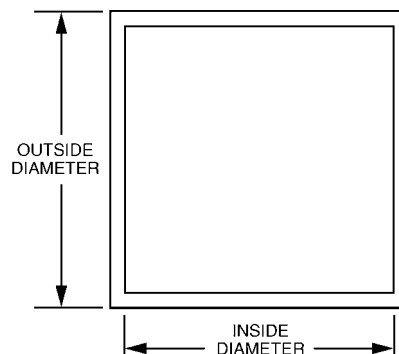
Figure 23 Square DONUT Primitive
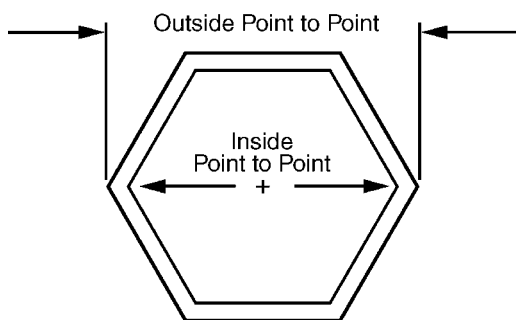
Figure 24 Hexagon DONUT Primitive
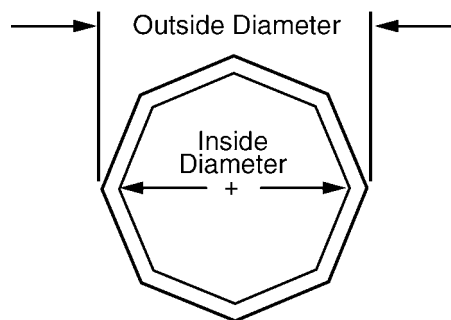
Figure 25   Octagonal DONUT Primitive

Example:

```
<DonutDef id="bd1:land12" shape="ROUND" outerDiameter="12.5"
          innerDiameter="12.0"/>
```

4.14.12 Thermal

The purpose of a thermal is to remove material from a plane or conductive filled area. To accomplish this the `Thermal` primitive shape differs from the other primitive shapes in that the default values are `PaintDesc/@type="VOID"` and `LineDesc/@type="ERASE"`. The `Thermal` shapes include square, round 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 non-functional lands on an interlayer plane, where the land is not connected to the plane. GenCAM defines these using the `Thermal` element with a spoke count of zero.

The examples shown in Figures 26 through 31 illustrate standard thermal primitive definitions. Many thermal primitive configurations can be generated using different spoke numbers and end types. The table of examples has been arranged with spokeless versions of the each shape in the right column.

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a qualifiedName that uniquely identifies the `Thermal` within the GenCAM file. The `id` is referenced by `thermalRef` attributes in order to create an instance of the thermal shape.
shape	REQUIRED	The shape of the thermal. One of ROUND | SQUARE | OCTAGON.
		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.
		OCTAGON – The inner and outer shapes are like `OctagonDef`.
outerDiameter	REQUIRED	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.
		OCTAGON – The point-to-point measurement on the x-axis of the octagon that forms the outer boundary of the thermal.

innerDiameter	REQUIRED	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.
		OCTAGON – The point-to-point measurement on the x-axis of the octagon that forms the inner boundary of the thermal.
spokeCount	DEFAULT	The number of cutouts allowed in the inner and outer shapes.
		ROUND – must be 0, 2, 3, or 4
		SQUARE – must be 0, 2, or 4
		OCTAGON – must be 0, 2, or 4
		The default value is 0.
		If the spokeCount is not defined (zero), the other three optional parameters do not apply. The spokeless thermal has a shape like a donut shape, but acts as a thermal by removing material.
spokeWidth	OPTIONAL	The minimum distance between the sides of a spoke cut. The default value is the innerDiameter subtracted from the outerDiameter
spokeStartAngle	OPTIONAL	The angle in counterclockwise direction from the x-axis at which the first spoke is cut. The default angle is 45 Degrees counterclockwise from the x-axis.
spokeEndShape	DEFAULT	The shape applied to the end of each spoke. One of ROUND, SQUARE, or PARALLEL. The default spokeEndShape is SQUARE.
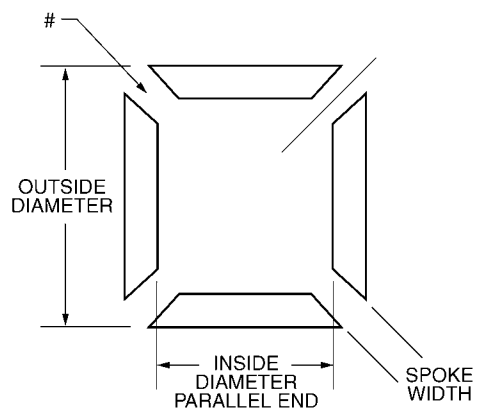
Figure 26 Square THERMAL Primitive
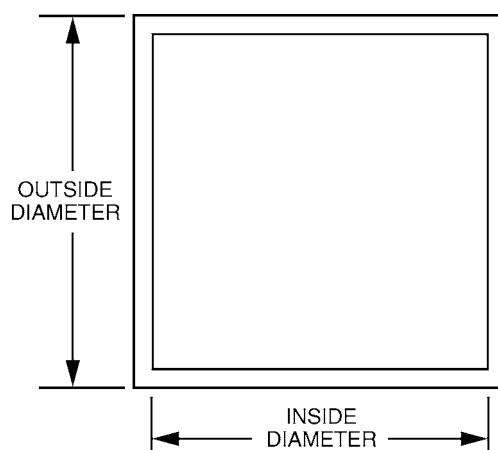
Figure 27 Square Spokeless THERMAL
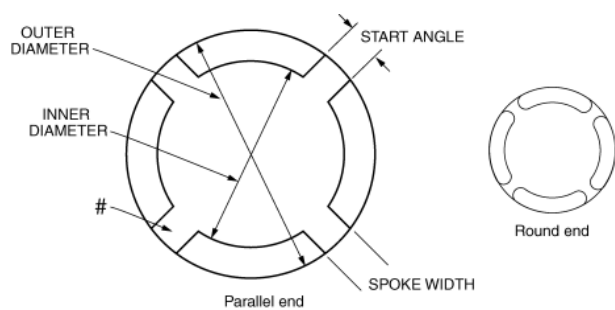
Figure 28 Round THERMAL Primitive
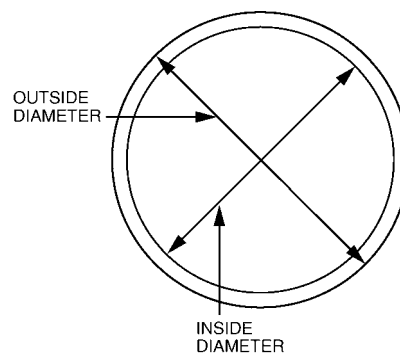
Figure 29 Round Spokeless THERMAL
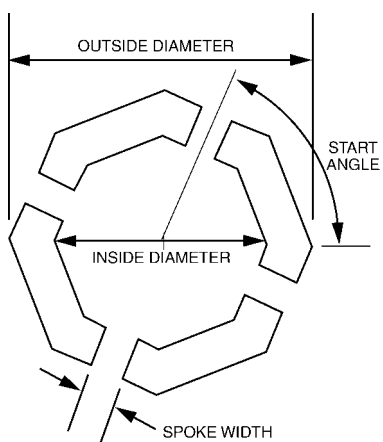
Figure 30 Octagonal THERMAL Primitive
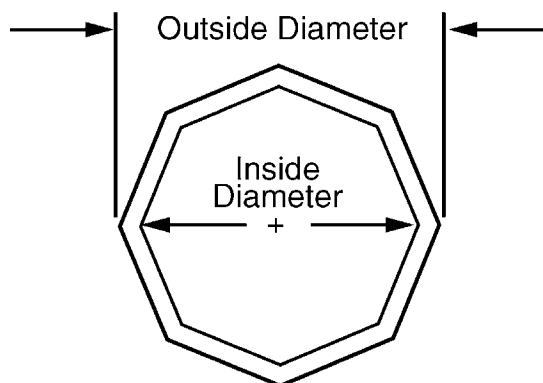
Figure 31 Octagonal Spokeless THERMAL

Examples:

```
<Thermal id="bd1:land12" shape="ROUND" innerDiameter="12.5"
            outerDiameter="12.0" />
```

4.14.13 PolygonDef

The `PolygonDef` element defines a sequence of connected edges that form a polygon. An edge can be straight, elliptical, or circular. The polygon is a two-dimensional, closed shape whose edges do not cross. The coordinates of the polygon are defined relative to the local coordinate system of the polygon. When a `PolygonRef` is used to create an instance of a polygon defined by a `PolygonDef` all of the properties of the `lineDescRef` in the `PolylineRef` apply to all line segments of the polygon. The properties of a `paintDescRef` attribute in a `PolygonRef` apply to the area enclosed inside the polygon. The attributes and children elements of a `PolygonDef` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the PolygonDef within the GenCAM file. The id is referenced by polygonRef attributes in order to create an instance of the polygon shape.
startXY	REQUIRED	The starting point of the polygon.

The `LineTo` element defines an edge to be drawn in the polygon.

Attributes	Requirement	Description
endXY	REQUIRED	The ending point of a straight line to be drawn from the previous point in the polygon.

59

The `CircArcTo` element defines an edge to be drawn in a polygon.

Attributes	Requirement	Description
center	REQUIRED	The center point for the circular arc to be drawn between the previous point and the `endXY`.
direction	DEFAULT	The direction to draw the circular arc between the previous point and the `endXY` about the center. One of CCLKW | CLKW. The default value is CCLKW.
endXY	REQUIRED	The end point of the circular arc to be drawn from the previous point in the polygon.

The `EllipArcTo` element defines an edge to be drawn in a polygon.

Attributes	Requirement	Description
focus1	REQUIRED	The first foci for the elliptical arc to be drawn between the previous point and the `endXY`.
focus2	REQUIRED	The second foci for the elliptical arc to be drawn between the previous point and the `endXY`.
direction	DEFAULT	The direction to draw the elliptical arc between the previous point and the `endXY` about focus1 and focus2. One of CCLKW | CLKW. The default value is CCLKW
endXY	REQUIRED	The end point of the circular arc to be drawn from the previous point in the polygon.

The `EndLine` element defines the final edge to be drawn in a polygon. The edge is to be drawn from the previous point in the polygon as a straight line to the `startXY` attribute for the polygon. There are no attributes for an `EndLine` element.

Attributes	Requirement	Description

The `EndCircArc` element defines the final edge to be drawn in a polygon. The edge is to be drawn from the previous point in the polygon as a circular arc to the startXY attribute for the polygon.

60

Attributes	Requirement	Description
center	REQUIRED	The center point for the circular arc to be drawn between the previous point and the startXY.
direction	DEFAULT	The direction to draw the circular arc between the previous point and the startXY about the center. One of CCLKW | CLKW. The default value is CCLKW.

The EndEllipArc element defines the final edge to be drawn in a polygon. The edge is to be drawn from the previous point in the polygon as a elliptical arc to the startXY attribute for the polygon.

Attributes	Requirement	Description
focus1	REQUIRED	The first foci for the elliptical arc to be drawn between the previous point and the startXY.
focus2	REQUIRED	The second foci for the elliptical arc to be drawn between the previous point and the startXY.
direction	DEFAULT	The direction to draw the circular arc between the previous point and the startXY about the center. One of CCLKW | CLKW. The default value is CCLKW.

Polygons may define the absence or presence of material. Figure 32 shows the characteristics for two polygons that have different paintdesc characteristics.
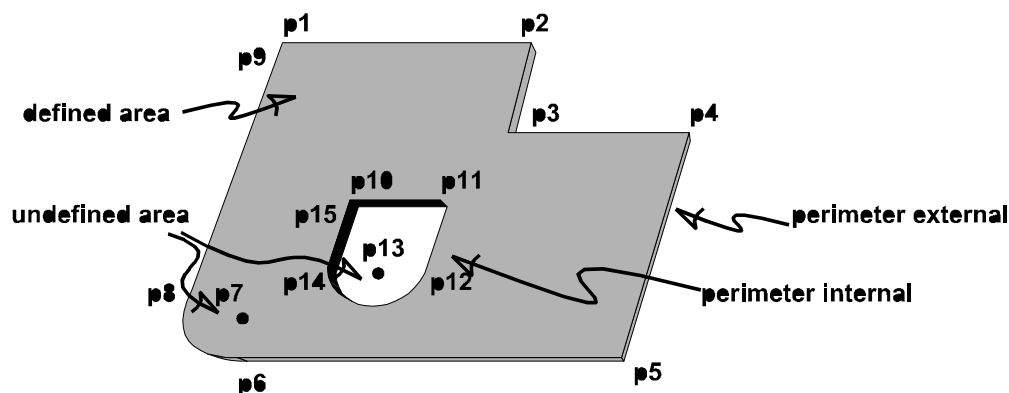
Figure 32 Example polygon used to define solder mask layer

4.14.14 PolylineDef

The PolylineDef element defines a sequence of connected edges that define a polyline. An edge can be straight, elliptical, or circular. The coordinates of the polyline are defined relative to the local coordinate system of the polyline. When a PolylineRef is used to create an instance of a polyline defined by a PolylineDef, all of the properties of the lineDescRef attribute in the PolylineRef apply to all line segments of a polyline. The attributes and children elements of a PolylineDef element are defined as follows:
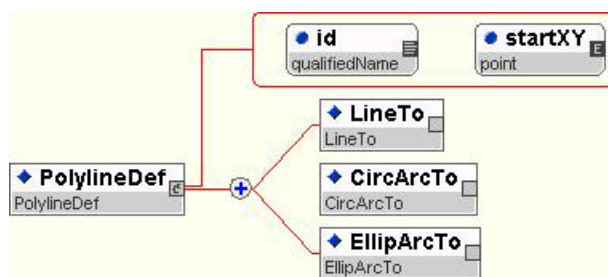
PolylineDef

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the PolylineDef within the GenCAM file. The id is referenced by polylineRef attributes in order to create an instance of the polyline shape.
startXY	REQUIRED	The starting point of the polyline.

The LineTo element defines an edge to be drawn in the polyline.

Attributes	Requirement	Description
endXY	REQUIRED	The ending point of a straight line to be drawn from the previous point in the polyline.

The CircArcTo element defines an edge to be drawn in a polyline.

Attributes	Requirement	Description
center	REQUIRED	The center point for the circular arc to be drawn between the previous point and the endXY.
direction	DEFAULT	The direction to draw the circular arc between the previous point and the endXY about the center. One of CCLKW	CLKW. The default value is CCLKW.
endXY	REQUIRED	The end point of the circular arc to be drawn from the previous point in the polyline.

The `EllipArcTo` element defines an edge to be drawn in a polyline.

Attributes	Requirement	Description
focus1	REQUIRED	The first foci for the elliptical arc to be drawn between the previous point and the `endXY`.
focus2	REQUIRED	The second foci for the elliptical arc to be drawn between the previous point and the `endXY`.
direction	DEFAULT	The direction to draw the elliptical arc between the previous point and the `endXY` about focus1 and focus2. One of CCLKW | CLKW. The default value is CCLKW
endXY	REQUIRED	The end point of the circular arc to be drawn from the previous point in the polyline.

Example of a Polyline
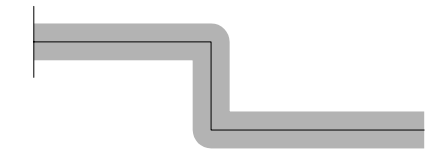
Figure 33 Three segment line description

63

4.15   PolygonBuilder

The `PolygonBuilder` element is an abstract element.   It can never be instanced. When `PolygonBuilder` is used the intention is to have one of the defined substitutionElements [include URL reference] appear in place of the `PolygonBuilder` element. The following table defines the substitution elements that are defined for the `PolygonBuilder` element.

PolygonBuilder Substitution Elements

PolygonRef

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
polygonRef	REQUIRED	A reference to a `PolygonDef/@id` attribute.

Polygon

The definitions of the children elements of `Polygon` are the same as the definitions used in the definition of `PolygonDef`. These definitions of the elements and attributes are defined in 4.14.13.

4.16   ClosedShape

The `ClosedShape` element is an abstract element.   It can never be instanced. When `ClosedShape` is used the intention is to have one of the defined substitutionElements [include URL reference] appear in place of the `ClosedShape` element. Instances of substitutionElements of a ClosedShape are drawn relative to the point of origin of their parent element. The following table defines the substitution elements that are defined for the `ClosedShape` element.

ClosedShape Substitution Elements		
Circle		
Attributes	**Requirement**	**Description**
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
diameter	REQUIRED	The diameter of the circle.
CircleRef		
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
circleRef	REQUIRED	A reference to a `CircleDef/@id` attribute.
RectCenter		
Attributes	**Requirement**	**Description**
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.

colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
width	REQUIRED	The length of the rectangle about the x-axis.
height	REQUIRED	The length of the rectangle about the y-axis.

RectCenterRef

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
rectCenterRef	REQUIRED	A reference to a `RectCenterDef/@id` attribute.

RectCham

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
width	REQUIRED	The length of the rectangle about the x-axis.
height	REQUIRED	The length of the rectangle about the y-axis.
chamfer	REQUIRED	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.

RectChamRef

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
rectChamRef	REQUIRED	A reference to a `RectChamDef/@id` attribute.

RectRound

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
width	REQUIRED	The length of the rectangle about the x-axis.
height	REQUIRED	The length of the rectangle about the y-axis.
radius	REQUIRED	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.

RectRoundRef

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
rectRoundRef	REQUIRED	A reference to a `RectRoundDef/@id` attribute.

RectCorner

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
p1	REQUIRED	The point that defines the lower left corner of the rectangle.
p2	REQUIRED	The point that defines the upper right corner of the rectangle.

RectCornerRef

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
rectCornerRef	REQUIRED	A reference to a `RectCornerDef/@id` attribute.

DShape

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.

endShape	REQUIRED	One of ROUND | FILLET | CHAMFER. Defines the type of modification that will be made to the corners on the right side of the rectangle.
		ROUND – trim back the right side of the rectangle to be a semicircle with a radius equal to ½ the width of the rectangle.
		FILLET – a radius is cut on the right side corners of the rectangle.
		CHAMFER – the top two corners are clipped at a 45 degree angle.
width	REQUIRED	The length of the rectangle about the x-axis.
height	REQUIRED	The length of the rectangle about the y-axis.
corner	OPTIONAL	the meaning if the endShape is:
		FILLET – the radius to be trimmed from the right side corners of the rectangle. If the radius is greater than ½ the height or ½ the width then the definition is undefined.
		CHAMFER – the length along the width and height of each corner between which the rectangle is to be clipped. If the value of chamfer is greater than ½ the height or ½ the width then the definition is undefined.

DShapeRef

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a LineDesc/@id attribute
colorRef	OPTIONAL	A reference to a Color/@id attribute. The default color is black.
paintDescRef	OPTIONAL	A reference to a PaintDesc/@id attribute.
dShapeRef	REQUIRED	A reference to a DshapeDef/@id attribute.

Oval

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a LineDesc/@id attribute
paintDescRef	OPTIONAL	A reference to a PaintDesc/@id attribute.

colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
width	REQUIRED	The length of the rectangle about the x-axis.
height	REQUIRED	The length of the rectangle about the y-axis. It is an error to define a height greater than the width.

OvalRef

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
ovalRef	REQUIRED	A reference to a `OvalDef/@id` attribute.

Diamond

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
width	REQUIRED	The length of the diamond along, and centered on, the x-axis.
height	REQUIRED	The length of the diamond along, and centered on, the y-axis.

DiamondRef

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
diamondRef	REQUIRED	A reference to a `DiamondDef/@id` attribute.

Hexagon

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
pointToPoint	REQUIRED	The distance between the two corner points of the hexagon on the x-axis.

HexagonRef

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
hexagonRef	REQUIRED	A reference to a `HexagonDef/@id` attribute.

Octagon

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
pointToPoint	REQUIRED	The distance between the two corner points of the octagon on the x-axis.

OctagonRef

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
octagonRef	REQUIRED	A reference to an `OctagonDef/@id` attribute.

All elements from the `PolygonBuilder` substitution group can be substituted wherever a `ClosedShape` element appears.

4.17   PolylineBuilder

The `PolylineBuilder` element is an abstract element. It can never be instanced. When `PolylineBuilder` is used the intention is to have one of the defined substitutionElements [include URL reference] appear in place of the `PolylineBuilder` element. Instances of substitutionElements of a `PolylineBuilder` are drawn relative to the point of origin of their parent element. The following table defines the substitution elements that are defined for the `PolylineBuilder` element.

<table>
<tr><td colspan="3" align="center">**PolylineBuilder Substitution Elements**</td></tr>
<tr><td colspan="3">

</td></tr>
<tr><td colspan="3" align="center">**PolylineRef**</td></tr>
<tr><td align="center">**Attributes**</td><td align="center">**Requirement**</td><td align="center">**Description**</td></tr>
<tr><td>lineDescRef</td><td>OPTIONAL</td><td>A references a LineDesc/@id attribute</td></tr>
<tr><td>colorRef</td><td>OPTIONAL</td><td>A reference to a Color/@id attribute. The default color is black.</td></tr>
<tr><td>polylineRef</td><td>REQUIRED</td><td>A reference to a PolylineDef/@id attribute.</td></tr>
<tr><td colspan="3">

Polyline

The definitions of the children elements of Polyline are the same as the definitions used in the definition of PolylineDef. These definitions for the elements and attributes are defined in 4.14.14.

</td></tr>
</table>

4.18   ShapeBuilder

The ShapeBuilder element is an abstract element. It can never be instanced. When ShapeBuilder is used the intention is to have one of the defined substitutionElements [include URL reference] appear in place of the ShapeBuilder element. Instances of substitutionElements of a ShapeBuilder are drawn relative to the point of origin of their parent element. The following table defines the substitution elements that are defined for the ShapeBuilder element.

73

ShapeBuilder Substitution Elements

Line
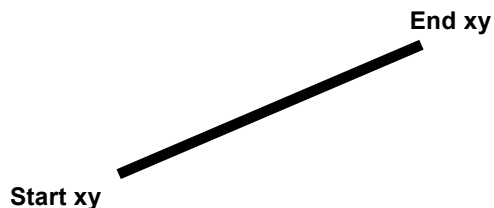
End xy

Start xy

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
startXY	REQUIRED	The beginning point of a circular arc.
endXY	REQUIRED	the ending point of a circular arc.

CircArc

End XY

Center → Start XY

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
startXY	REQUIRED	The beginning point of a circular arc.
endXY	REQUIRED	The end point of the circular arc.
center	REQUIRED	The center point for the circular arc to be drawn between the `startXY` and the `endXY`.

direction	DEFAULT	The direction to draw the circular arc between the `startXY` point and the `endXY` about the center. One of CCLKW	CLKW. The default value is CCLKW.

EllipArc

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
startXY	REQUIRED	The beginning point of a elliptical arc.
endXY	REQUIRED	The end point of the elliptical arc.
focus1	REQUIRED	The first foci for the elliptical arc to be drawn between `startXY` and the `endXY`.
focus2	REQUIRED	The second foci for the elliptical arc to be drawn between `startXY` and the `endXY`.
direction	DEFAULT	The direction to draw the elliptical arc between the `startXY` point and the `endXY` about the foci. One of CCLKW	CLKW. The default value is CCLKW.

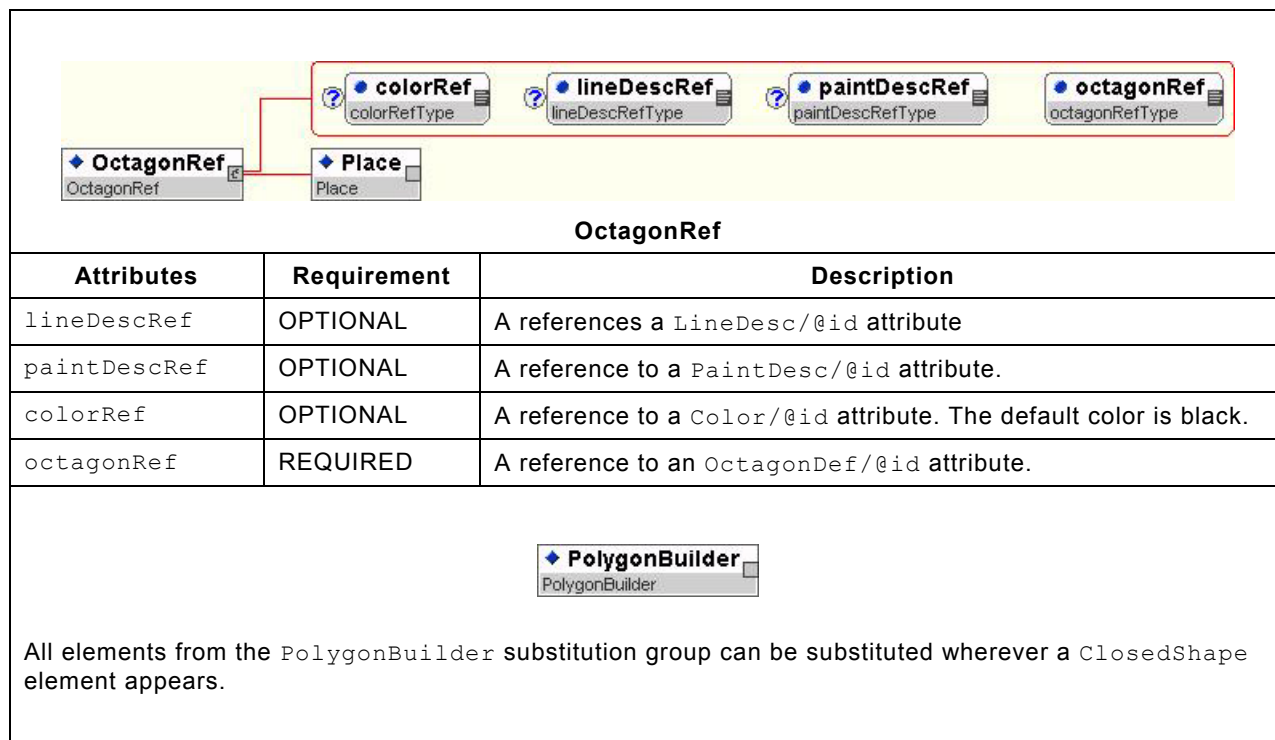**Donut**

Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.

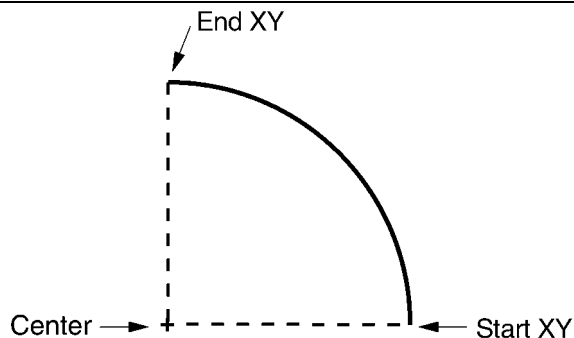| shape	REQUIRED	The inner and outer shapes are one of ROUND, SQUARE , HEXAGON or OCTAGON:
		ROUND `circleDef.` – The inner and outer shapes are like
		SQUARE – The inner and outer shapes are like `RectCenterDef` with height and width of each shape being equal.
		HEXAGON – The inner and outer shapes are like `HexagonDef.`
		OCTAGON – The inner and outer shapes are like `OctagonDef.`
outerDiameter	REQUIRED	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.
innerDiameter	REQUIRED	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.

76

DonutRef
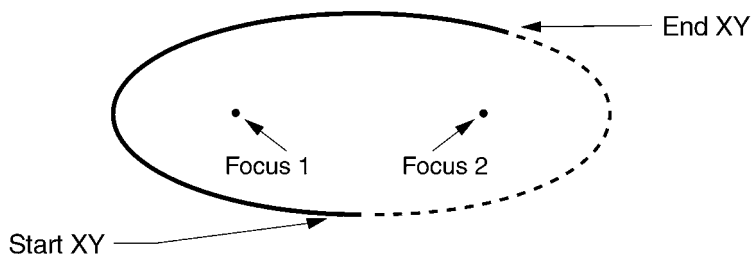
Attributes	Requirement	Description
lineDescRef	OPTIONAL	A references a `LineDesc/@id` attribute
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
paintDescRef	OPTIONAL	A reference to a `PaintDesc/@id` attribute.
donutRef	REQUIRED	A reference to a `DonutDef/@id` attribute.

All elements from the `ClosedShape` substitution group can be substituted wherever a `ShapeBuilder` element appears.

All elements from the `PolylineBuilder` substitution group can be substituted wherever a `ClosedShape` element appears.

4.19   Features

The `Features` element defines a list of `Feature` elements that are referenced by other elements in the GenCAM file. A variety of features exist that are used in printed boards, printed board drawings, and printed board assembly. Some of these relate to the characteristic for defining polarity of electronic components. A special `Feature` element is available for defining the artwork necessary to draw these special symbols. The `Feature` elements are referenced by `FeatureRef` element or an Image element when an instance of a feature is to be drawn on a product or on a drawing.

4.19.1   Feature

The `Feature` element defines a feature artwork. The feature provide a visual reference, e.g., a diode symbol to indicate the polarity of the diode when placed on the board. A feature is a collection of `ShapeBuilder` elements that are drawn relative to a common origin. The attributes of `Feature` are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the Feature within the GenCAM file. The id is referenced by featureRef attributes in order to create an instance of the feature.
lineDescRef	OPTIONAL	A references a LineDesc/@id attribute
colorRef	OPTIONAL	A reference to a Color/@id attribute. The default color is black.
paintDescRef	OPTIONAL	A reference to a PaintDesc/@id attribute.

4.19.2   Feature Examples

The following examples illustrate how Feature can be used to create reusable features that can be instanced in a product.

4.19.2.1 Feature (Plus-sign)

The following example defines a `Feature` in the shape of a plus-sign by drawing a single polygon that traces the perimeter of the plus-sign shape:

Position Datum

```
<Feature id = "bd1:plus_sign1" paintDesc = "IPCstd:blackFill">
      <Polygon startXY = "-1 -4">
            <LineTo endXY = "-1 -1"/>
            <LineTo endXY = "-4 -1"/>
            <LineTo endXY = "-4 1"/>
            <LineTo endXY = "-1 1"/>
            <LineTo endXY = "-1 4"/>
            <LineTo endXY = "1 4"/>
            <LineTo endXY = "1 1"/>
            <LineTo endXY = "4 1"/>
            <LineTo endXY = "4 -1"/>
            <LineTo endXY = "1 -1"/>
            <LineTo endXY = "-1 -4"/>
            <EndLine/>
      </Polygon>
</Feature>
```
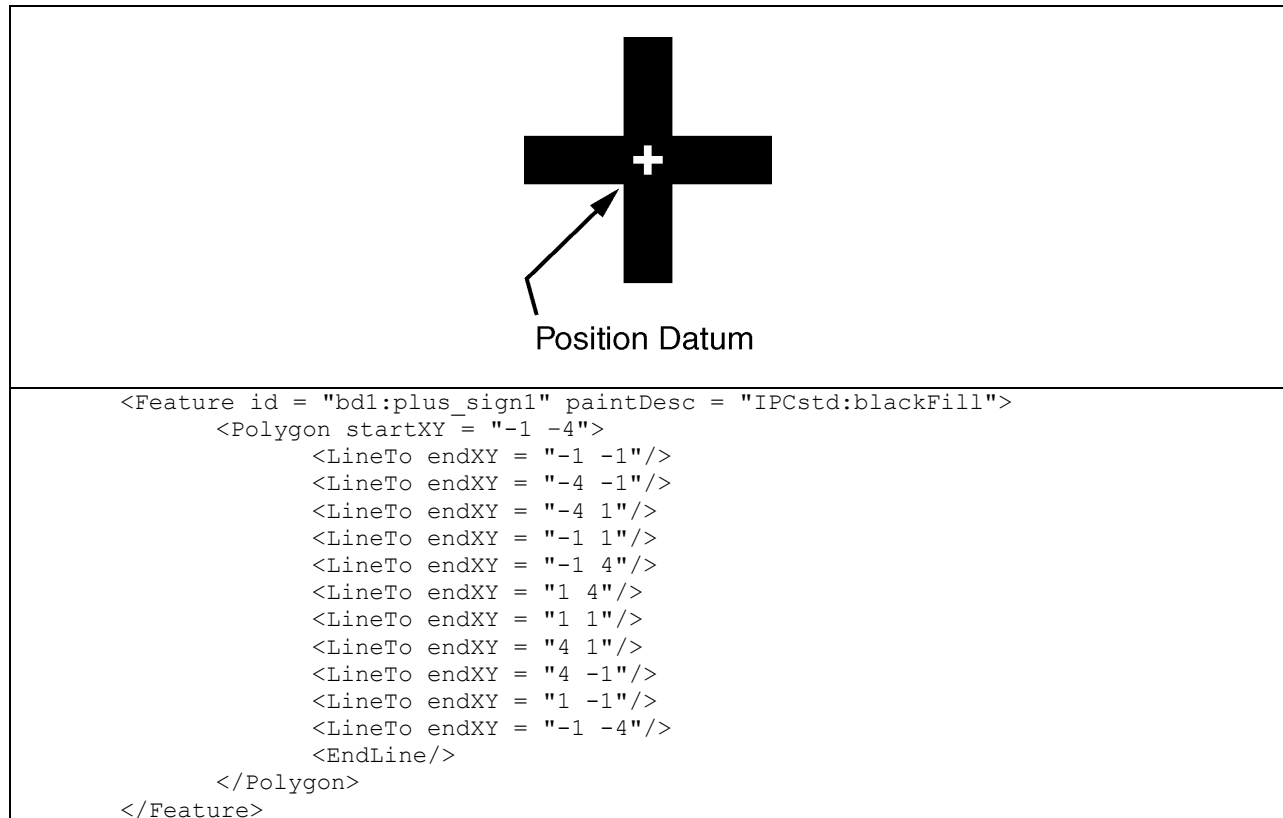
Figure 34 Crosshair Artwork Feature

Note: The white cross in the center of the crosshair is not actually drawn. It is present to indicate the location of the point of origin.

The crosshair `Feature` could also be defined using two overlaping rectangles. The following example implements Figure 35 a `Feature` named "`bd1:plus_sign2`" that is drawn as two rectangles sharing a common center.

Position Datum

```
<Feature id = "bd1:plus_sign2" colorRef = "IPCstd:black"
            paintDescRef = "IPCstd:Filled">
    <RectCenter width = "4" height = "2">
        <Place x = "0" y = "0"/>
    </RectCenter>
    <RectCenter width = "2" height = "4">
        <Place x = "0" y = "0"/>
    </RectCenter>
</Feature>
```
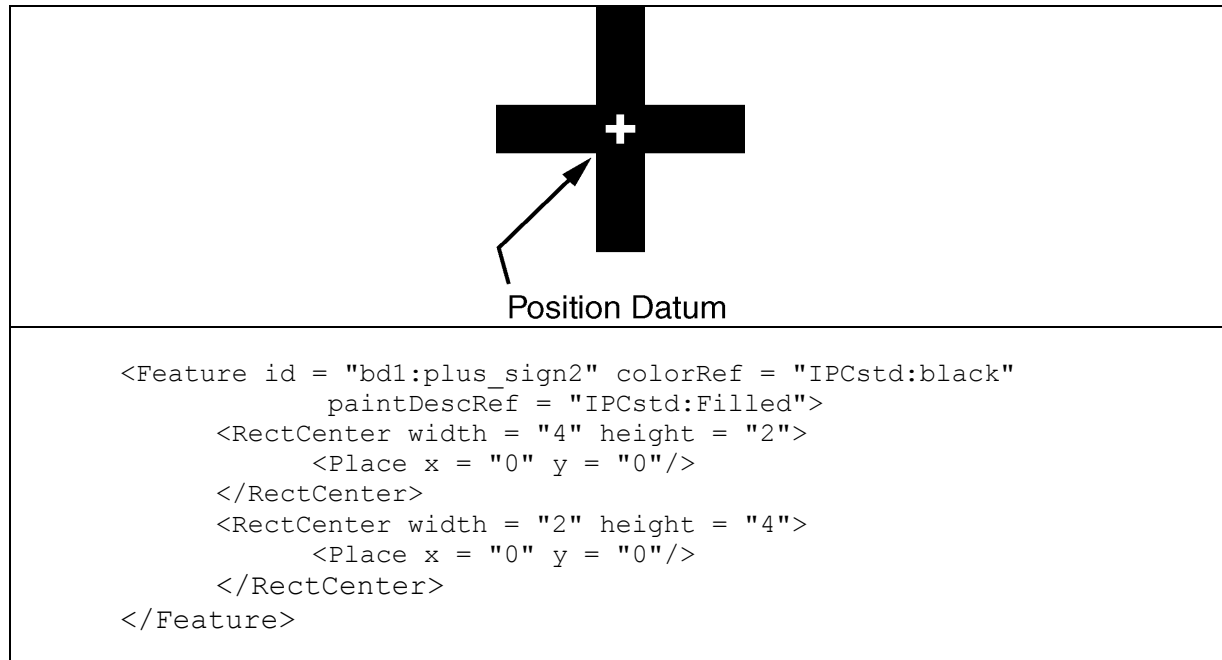
Figure 35   Crosshair Artwork using RecCenter

4.19.2.2 Feature (Diode)

This example illustrates the use of a `Feature` shape to create a diode symbol. The shape is constructed by drawing a triangle with the apex of the triangle touching the center of a vertical line.


```
        <Feature id = "bd1:diode7" colorRef = "IPCstd:Black"
                 paintDescRef = "IPCstd:filled">
            <Polygon startXY = "0 0">
                <LineTo endXY = "4 2"/>
                <LineTo endXY = "4 -2"/>
                <EndLine/>
            </Polygon>
            <RectCenter width = "1" height = "4">
                <Place x = "-0.5" y = "0.0"/>
            </RectCenter>
        </Feature>
```
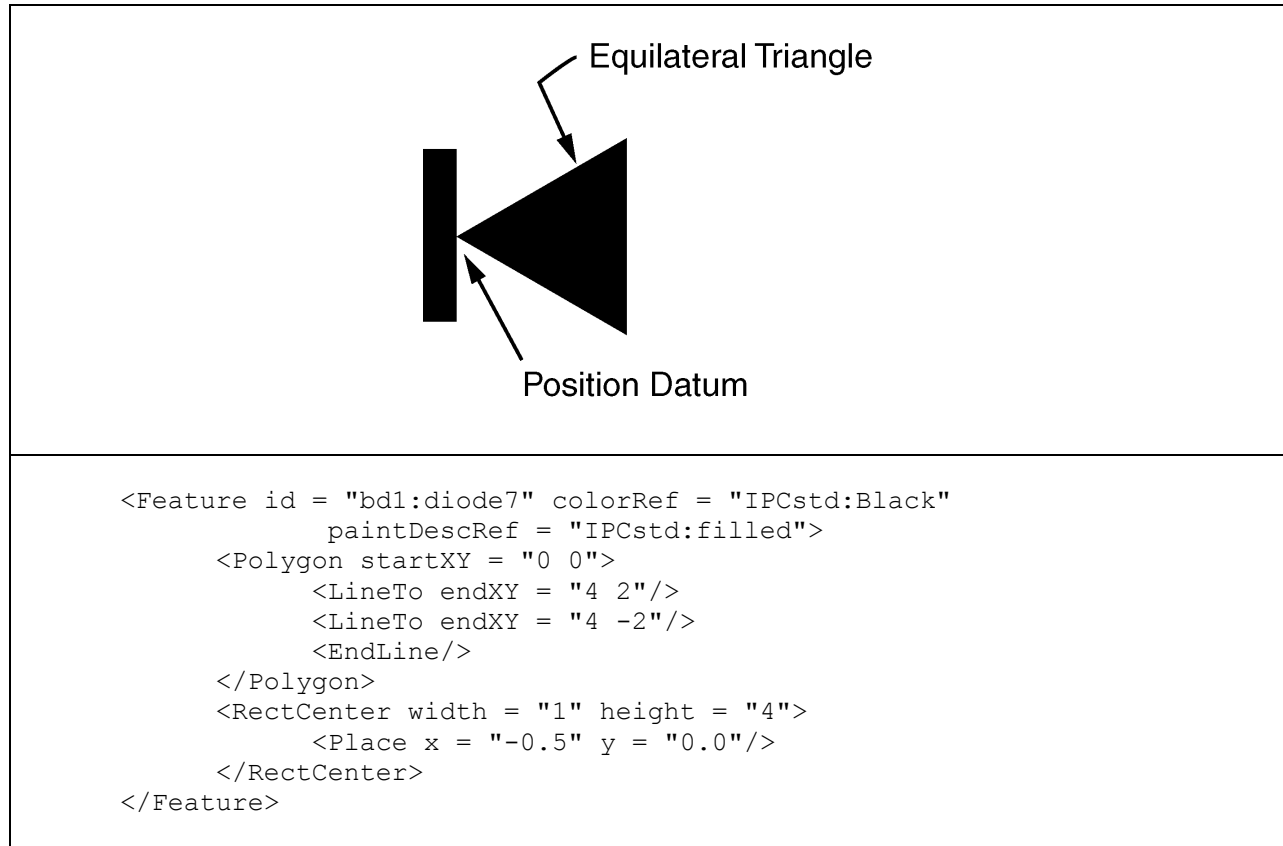
4.20   Targets

The `Targets` element defines a list of `Target` elements that are referenced by other elements in the GenCAM file. Targets are used in printed board descriptions as patterns that assist in registration of layers to each other, that aid in the alignment of boards/panels to tooling positions, and that mark significant information on the part, drawing, board, panel or assembly. The `Target` elements are referenced by `TargetRef` element when an instance of a feature is to be drawn on a product or on a drawing.

4.20.1   Target

The Target element defines a target artwork. A target is used for registration and alignment during the manufacturing process. A target is a collection of ShapeBuilder elements that are drawn relative to a common origin. The attributes of a Target element are defined as follows:
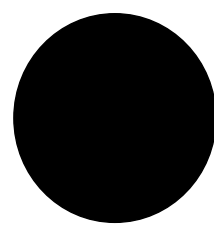
Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the Target within the GenCAM file. The id is referenced by TargetRef element in order to create an instance of the target.
function	REQUIRED	One of REGISTRATION | ALIGNMENT | MARKER.
colorRef	OPTIONAL	A reference to a Color/@id attribute. The default color is black.

4.20.2   Target Examples
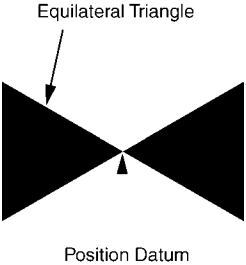
Targets are artworks that perform a special purpose. The following examples illustrate how targets are to be defined in a GenCAM file.

4.20.2.1   ALIGNMENT (Fiducials)

Fiducials are solid, typically round images defined by a diameter. The GenCAM Target can be used to create a fiducial. The following example demonstrates a fiducial that is a solid filled circle.


```
<Target id = "IPCstd:fiducial" function = "ALIGNMENT"
          colorRef = "IPCstd:black">
     <Circle diameter = "2.0" paintDescRef = "IPCstd:compfill">
          <Place x = "0" y = "0"/>
     </Circle>
</Target>
```

A butterfly target is useful as an alignment feature in phototool imaging. The following example illustrates a butterfly fiducial for GenCAM using the `Target` element and the reserved word ALIGNMENT.

Equilateral Triangle

Position Datum
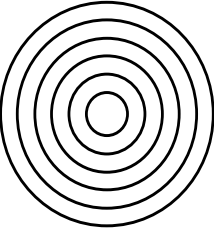
```
<Target id = "IPCstd:fiducial" function = "ALIGNMENT"
        colorRef =  "IPCstd:black" >
    <Polygon startXY = "0 0">
        <LineTo endXY = "4 2"/>
        <LineTo endXY = "4 -2"/>
        <EndLine/>
    </Polygon>
    <Polygon startXY = "0 0">
        <LineTo endXY = "-4 2"/>
        <LineTo endXY = "-4 -2"/>
        <EndLine/>
    </Polygon>
</Target>
```

4.20.2.2   REGISTRATION (Moiré)

Moiré patterns are used to register one layer to another. Registration of this sort is very important to the creation of effective phototools. GenCAM uses the `Target` element a "function" type REGISTRATION to define registration shapes. The following example illustrates the definition of a typical Moiré REGISTRATION target.

```
<Target id = "IPCstd:moireTarget" function = "REGISTRATION">
    <Circle diameter = "4.5" lineDescRef ="IPCstd:targetLines">
        <Place x = "0" y = "0"/>
    </Circle>
    <Circle diameter = "4.0" lineDescRef ="IPCstd:targetLines">
        <Place x = "0" y = "0"/>
    </Circle>
    <Circle diameter = "3.5" lineDescRef ="IPCstd:targetLines">
```

84

```
                <Place x = "0" y = "0"/>
        </Circle>
        <Circle diameter = "3.0" lineDescRef ="IPCstd:targetLines">
                <Place x = "0" y = "0"/>
        </Circle>
        <Circle diameter = "2.5" lineDescRef ="IPCstd:targetLines">
                <Place x = "0" y = "0"/>
        </Circle>
        <Circle diameter = "2.0" lineDescRef ="IPCstd:targetLines">
                <Place x = "0" y = "0"/>
        </Circle>
</Target>
```

Figure 36 Moire Artwork

4.20.2.3 MARKER (Bullseye)

The bullseye may also be used as a marker in many applications. The GenCAM Target element and the reserved word MARKER can be used to demonstrate this use case.

Figure 36 Moire Artwork

4.20.2.3   MARKER (Bullseye)

The bullseye may also be used as a marker in many applications. The GenCAM Target element and the reserved word MARKER can be used to demonstrate this use case.

Dot diameter

Inner diameter

Outer diameter

```
<Target id = "IPCstd:bullseye" function = "MARKER">
      <Circle diameter = "4.5 " paintDescRef = "IPCstd:filled"
            colorRef = "IPCstd:black">
          <Place x = "0" y = "0"/>
      </Circle>
      <Circle diameter = "4.0 " paintDescRef = "IPCstd:void"
            colorRef = "IPCstd:black">
          <Place x = "0" y = "0"/>
      </Circle>
      <Circle diameter = "1.5 " paintDescRef = "IPCstd:filled"
            colorRef = "IPCstd:black">
          <Place x = "0" y = "0"/>
      </Circle>
</Target>
```
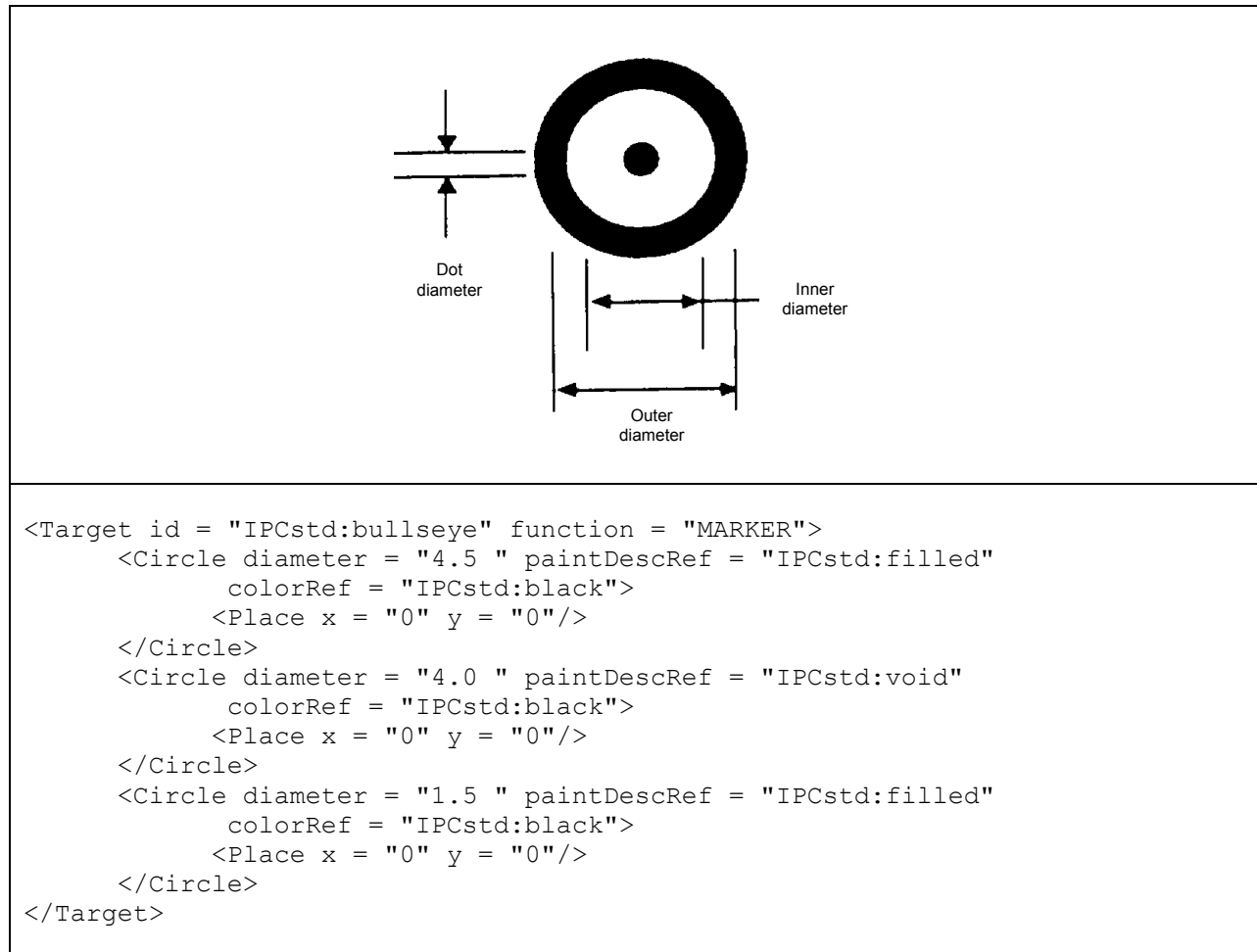
Figure 37 Bullseye ARTWORK

4.21   Fonts

The `Fonts` element defines a list of all font types that are used within the GenCAM file. The list of `Fonts` in a GenCAM can include user-defined font (`Ufont`) as well as references to industry standard `Font` definitions. The default font for GenCAM is of font type *Helvetica*.
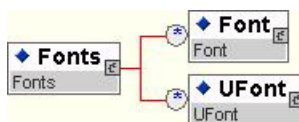
4.21.1   Font

When an industry standard font is specified in a GenCAM file the font type is assigned a name The following describes the elements and attributes used for defining a Font element:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the Font as a Font type that can be referenced within the GenCAM file. The `id` is referenced by `fontRef` attribute of a Text element. The font type defined the font used to render the Text.
url	REQUIRED	`uri` reference (required when other than default is used)

4.21.1.1   User-defined character sets (UFONT)

The `Ufont` element can be used   for creating user-defined character sets. User defined character sets require each character be defined as a `Glyph`.

The bounding rectangle rules for Text apply when using UFont. The major difference is that since `Ufont` is one character at a time, the entire font is described using a character code for each symbol established as a `Glyph`. The `Glyph` character code, therefore, identifies the symbol or letter shape. The following tables define the requirements for `Ufont` and `Glyph` elements. The `Edit` element is as described in 3.1.1.

The `Glyph` element consists of a `ShapeBuilder` element used to describe the character. There are several attributes attached to the `Glyph` element.

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the Ufont as a Font type that can be referenced within the GenCAM file. The id is referenced by fontRef attribute of a Text element. The font type is used in rendering the.

Attributes	Requirement	Description
charCode	REQUIRED	An integer in the range of 0-255 that is used to select the Glyph from the Ufont character set. The charCode is the character code of this user-defined character.
p1	REQUIRED	A point defining the lower left hand corner of the character cell
p2	REQUIRED	A point defining the upper right hand corner of the character cell

The bounding rectangle of the character cell is defined by p1 and p2. The position of the next glyph in a bounding rectangle is defined by subtracting the x value of p1 from the x value of p2 and adding it to the p2 location of the current glyph. All shapes in a glyph are clipped at the boundaries of the character cell.

4.22   Logos

The Logos element defines a list of all the Logo elements used within the GenCAM file.
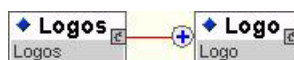
4.22.1   Logo

Logos are user-defined artworks that are to be used when incorporating a company logo or certification symbol into a drawing or printed board design. Some examples are shown in Figure 38:
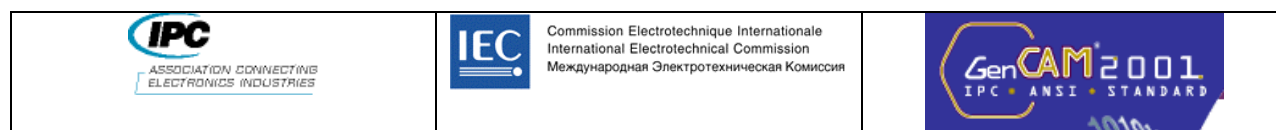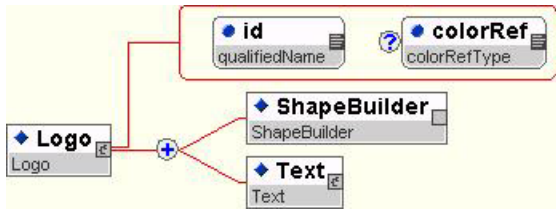
Figure 38. Examples of Logos
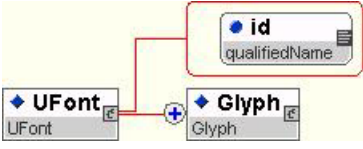
Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the Logo that can be referenced within the GenCAM file. The id is referenced by logoRef attribute of the LogoRef element.
colorRef	OPTIONAL	A reference to a Color/@id attribute. The default color is black.
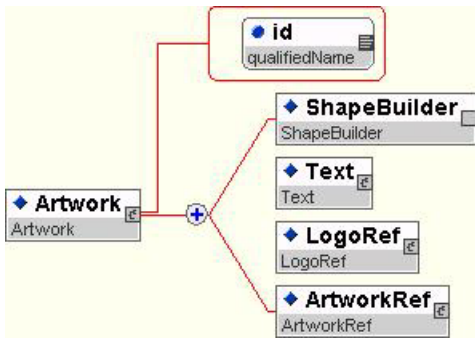
4.23   Artworks

The Artworks element defines a list of all the Artwork elements used within the GenCAM file.
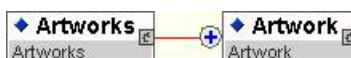
4.23.1   Artwork

The Artwork element defines an general purpose artwork. The artwork definition is referenced by elements as a reusable collection of graphics. An artwork is a collection of Text elements, ShapeBuilder elements, references to previously defined Artwork elements or previously defined Logo elements. The collection of drawable items are drawn relative to a common local point of origin. The attribute of an Artwork element and the children elements are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the Artwork within the GenCAM file. The id is referenced by artworkRef attributes in order to create an instance of the artwork.

Attributes	Requirement	Description
logoRef	REQUIRED	A reference to a `Logo/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.

Attributes	Requirement	Description
artworkRef	REQUIRED	A reference to an `Artwork/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
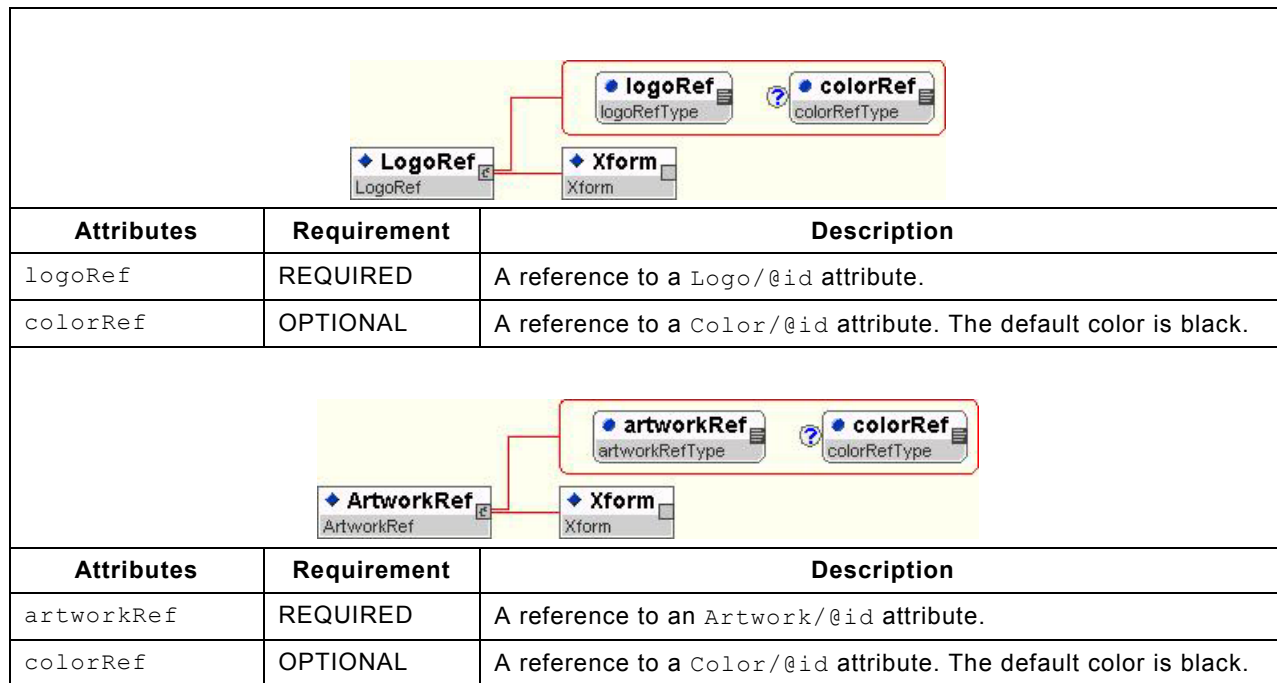
4.24   Layers

The `Layers` element defines a list of all the `LayersSingle`, `LayerSet`, and `LayerSwap` elements used within the GenCAM file. A layer can be used to manage or collect information about a physical layer (e.g. a bottom conductor layer) or an abstract layer (e.g. a layer containing package outlines) of the product. The following diagram defines the order and occurance constraint of elements in the `Layers` element:

90
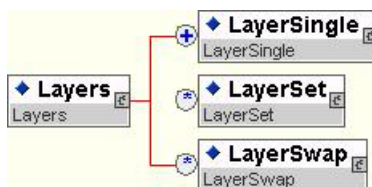
4.24.1   LayerSingle

A `LayerSingle` element defines the physical characteristics of a layer and assigns a GenCAM layer type to a layer. The attributes of the `LayerSingle` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `LayerSingle` element within the GenCAM file. The `id` is referenced by a `layerSingleRef` attribute or `layersRef` attribute in order to define which layers to which the referencing element applies.
surface	REQUIRED	A fixed field parameter that defines the surface of the layer. One of: TOP | BOTTOM | BOTH | INTERNAL | ALL | NOTAPPLICABLE
type	REQUIRED	The function of the layer. One of: LEGEND | GLUE | SOLDERMASK | BOARDOUTLINE | COATINGCOND | COATINGNONCOND | CONDUCTOR | COURTYARD | DIELBASE | DIELCORE | DIELPREG | DIELADHV | SOLDERBUMP | PASTEMASK | HOLEFILL | PIN | COMPONENT | RESISTIVE | CAPACITIVE | PROBE | REWORK | FIXTURE | GRAPHIC
material	OPTIONAL	The descriptive name for the type of material of the layer. The materials are defined in IPC-4101, IPC-4102, IPC-4103, IPC-4104.
materialCode	OPTIONAL	The material code of the layer defined in IPC-4101, IPC-4102, IPC-4103, IPC-4104 nomenclature. E.g. "L21 1500 C1/C1 BA" L – material designator. 21 – specification sheet number. 1500 – nominal laminate thickness. C1/C1 – metal cladding (type and thickness). B – thickness tolerance class. A – surface quality class.
thickness	OPTIONAL	The physical dimension of the layer.
profileTolRef	OPTIONAL	A reference to a `ProfileTolerance/@id` attribute.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
functionCategory	REQUIRED	A general category for the layer type. One of BOARDFAB | BOARDTEST | ASSEMBLY | ASSEMBLYTEST | FIXTURE | DISPLAY

4.24.2   LayerSet

The `LayerSet` element defines a collection (set) of layers that are referenced as a sequence or otherwise related grouping of `LayerSingle` and `LayerSet` elements. A layerset can be used for defining sequential lamination, cutouts, padstacks, vias, etc. The order of the layer references is significant, and must sequentially reflect the layer stack, as viewed from the primary component side of the board, inward. The attributes of a `LayerSet` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `LayerSet` element within the GenCAM file. The `id` is referenced by a `layersRef` attribute in order to define which layers to which the referencing element applies. If the CAD system does not define names for sets of layers then the GenCAM writer must supply names such as "`bd1:layerset2`".
thickness	OPTIONAL	The physical dimension of the `LayerSet` thickness.
profileTolRef	OPTIONAL	A reference to a `ProfileTolerance/@id` attribute. In this context the tolerance attributes `finishedLMC` and `finishedMMC` apply to the material thickness of the full `LayerSet`.

4.24.3   Layer

The `Layer` element reference a layer or set of layers that is to be included in a `LayerSet` definition. The only attribute of a `Layer` element is defined as follows:
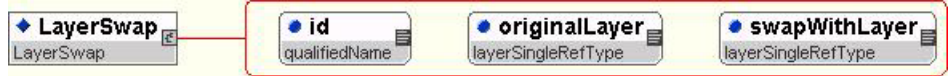
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of a layer that is to be included in this `LayerSet`.

4.24.4 PreferredVendors

◆ **PreferredVendor**	● **enterpriseRef**	● **personRef**
PreferredVendor	enterpriseRefType	personRefType

enterpriseRef	REQUIRED	A reference to a `Enterprse/@id` attribute that is the prefered vendor for the layerset.
personRef	OPTIONAL	The `personRef` attribute references the `Person/@id` to identify the contact point at the preferred vendor.

4.24.5 LayerSwap

A `LayersSwap` element defines a pair of layers that are to be swapped when the transformation is set to MIRROR instead of NOMIRROR. The attributes of a `LayerSwap` element are defined as follows:

◆ **LayerSwap**	● **id**	● **originalLayer**	● **swapWithLayer**
LayerSwap	qualifiedName	layerSingleRefType	layerSingleRefType

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `LayerSwap` element within the GenCAM file. This `id` is only referenced in a GenCAM file when an Edit element is needed. The `id` could be referenced if the `LayerSwap` is being edited.
originalLayer	REQUIRED	A reference to a `Layer/@id` attribute of a layer that is to swapped with `swapWithLayer`.
swapWithLayer	REQUIRED	A reference to a `Layer/@id` attribute of a layer that is to swapped with `originalLayer`.

4.25   Pads

The `Pads` element defines a list of all the `Pad` elements used within the GenCAM file.

4.25.1   Pad

The `Pad` element defines a pad shape for use in a padstack. The attributes of a `Pad` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `Pad` element within the GenCAM file. The `id` is referenced by a `padRef` attribute in order to create an instance of the pad.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
lineDescRef	OPTIONAL	A reference to a `LineDesc/@id` attribute.
paintDescRef	REQUIRED	A reference to a `PaintDesc/@id` attribute.
padPrimitiveRef	REQUIRED	A reference to the `id` attribute of one of the following element types: CircleDef | RectCenterDef | RectChamDef | RectRoundDef | OvalDef | DiamondDef | HexagonDef | OctagonDef | ThermalDef|DshapeDef (ThermalDef removes material. They are only applicable when drawn on a plane)
profileTolRef	OPTIONAL	A reference to a `ProfileTolerance/@id` attribute. The tolerance defined by the referenced `ProfileTolerance` applies to the tolerance allowed in the placement of the shape defined by the referenced pad.

4.26   BarrelDescs

The `BarrelDesc` element defines the characteristics used when plating, coating, and filling a cavity. The attributes of a `BarrelDesc` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a `qualifiedName` that uniquely identifies the `BarrelDesc` within the GenCAM file. The id is referenced by `BarrelDescRef` attributes in order to associate the `BarrelDesc` definition with geometries to which it applies.
fillMaterial	OPTIONAL	Identifies the material that is used to fill cavity of a HOLE, CUTOUT, SLOT, or WELL. If the attribute is omitted the cavity is not filled.

4.26.1   Barrel

The `Barrel` element describes the characteristics of one layer of plating or coating that is applied to a barrel or to a wall. For walls that require multiple plates, or coatings, the order of `Barrel` definitions is significant. The `Barrel` must be applied in the order in which they are defined in the `BarrelDesc`. The attributes of a `Barrel` element are defined as follows:

Attributes	Requirement	Description
barrelType	REQUIRED	The type of barrel is either COAT or PLATE.
material	DEFAULT	The coating or plating material for the `Barrel`. One of COPPER | CARBON | GRAPHITE | NICKELGOLD | TINLEAD | NICKEL | GOLD | PALADIUM | TIN | ELECTROLYTICCOPPER | ELECTROLESSCOPPER. The default is ELECTROLESSCOPPER.
minThickness	REQUIRED	The minimum thickness of material that would be within the limits set by the `ProfileDesc/@finishedLMC` attribute of the item being plated.
maxThickness	REQUIRED	The maximum thickness of the material is set by the `ProfileDesc/@finishedLMC` of the item being plated.

4.27   Holes

The `Holes` element defines a list of all the `Hole` elements used within the GenCAM file.

4.27.1   Hole

The Hole element defines a hole shape. The attributes of a Hole element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `Hole` element within the GenCAM file. The `id` is referenced by a `holeRef` attribute in order to create an instance of the hole.
type	REQUIRED	The type of hole is one of ELECTRICAL, MECHANICAL or TOOLING.
primitiveRef	REQUIRED	A reference to the `id` attribute of one of the following element types: CircleDef | RectCenterDef | RectChamDef | RectRoundDef | OvalDef | DiamondDef | HexagonDef | OctagonDef
barrelDescRef	OPTIONAL	A reference to a `barrelDesc/@id` attribute.
profileTolRef	OPTIONAL	A reference to a `ProfileTolerance/@id` attribute. The tolerance defined by the referenced `ProfileTolerance` applies to the tolerance allowed in the placement of the hole defined by the referenced `primitiveRef`.
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers through which the hole referenced by `padRef` is to be cut.
displayGraphics	OPTIONAL	A reference to an `Artwork/@id` A `displayGraphics` artwork is displayed by a CAD or CAM system instead of the real physical dimension of the hole. The purpose is to improve the distinction between holes that are difficult to differentiate in size by visual inspection.

4.28   PadStacks

The `PadStacks` element defines a list of all the `PadStack` elements used within the GenCAM file.

4.28.1   PadStack

A `PadStack` element must be defined for each padstack used in the GenCAM file. The only attribute of a `PadStack` element is defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `PadStack` element within the GenCAM file. The `id` is referenced by a `padStackRef` attribute in order to create an instance of the padstack.

4.28.1.1   PadRef

The `PadRef` element positions a pad in a padstack. The `padRef` attribute references a `Pad` element that is to be added to the `PadStack` on the layers referenced by `layersRef`. The attributes of a `PadRef` element are defined as follows:

Attributes	Requirement	Description
padRef	REQUIRED	A reference to a `Pad/@id` attribute. The pad defined by `Pad` is instanced relative to the local point of origin.
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers that the pad referenced by `padRef` is to be drawn.
locationTolRef	OPTIONAL	A reference to a `TruePositionTolerance/@id` attribute. The tolerance defined by the referenced `TruePositionTolerance` applies to the tolerance allowed in the placement of the shape defined by the referenced pad.

4.28.1.2   ArtWorkRef

The `Artwork`Ref element positions an artwork in a padstack. The `artworkRef` attribute references an `Artwork` element that is to be added to the `PadStack` on the layers referenced by `layersRef`. A `PadRef` should be used in preference to an `ArtworkRef`. Only use an `ArtworkRef` if the shape of the pad cannot be expressed using a shape defined with a `padPrimitiveRef`. The attributes of an `ArtworkRef` element are defined as follows:
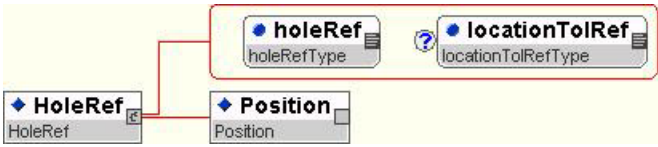
Attributes	Requirement	Description
artworkRef	REQUIRED	A reference to a `Artwork/@id` attribute. The pad defined by `Pad` is instanced relative to the local point of origin.
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers that the `Artwork` referenced by `artworkRef` is to be drawn.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.

4.28.1.3 HoleRef

The `HoleRef` element positions a drilled or punched feature. The `HoleRef` element references `Hole/@id` attribute to add an instance of the hole to the parent element. The layers to be drilled or punched by a `HoleRef` are defined by `Hole`. (By placing the layers reference in the Hole the process step, such as drilling hole for a buried via, are easier to identify.) The attributes of a `HoleRef` element are defined as follows:

Attributes	Requirement	Description
holeRef	REQUIRED	A reference to a `Hole/@id` attribute. The hole defined by `Hole` is instanced relative to the local point of origin.
locationTolRef	OPTIONAL	A reference to a `TruePositionTolerance/@id` attribute. The tolerance defined by the referenced `TruePositionTolerance` applies to the tolerance allowed in the placement of the hole defined by the referenced `Hole`.

4.29   Patterns

The `Patterns` section defines Pattern elements that are composed of reusable items, such as padstacks, artworks, targets, holes, and text. All items used in a pattern definition share a common origin. A pattern is constructed by referencing items defined in earlier section.

4.29.1   Pattern

The `Pattern` element defines a collection of items, such as pads, holes, and artworks that are grouped together for reuse. For instance, the definition of standard land patterns use to attach surface mount devices can be encoded in a `Pattern`. An engineer laying out a board will reduce the risk of error by using the IPC-782 definitions of standard land patterns. These patterns encode best practices design information and by referencing standard patterns when defining mounting locations on a board or panel the designer eliminates many potential design rule violations.

All items included in the pattern definition share a common point of origin. The information in a pattern defines the features associated with a mounting location that are fabricated on a board or a panel. A PatternRef in a MountingLocation places the pattern on a panel or a board. The attributes of a `Pattern` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `Pattern` element within the GenCAM file. The `id` is referenced by a `patternRef` attribute in order to create an instance of the pattern.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
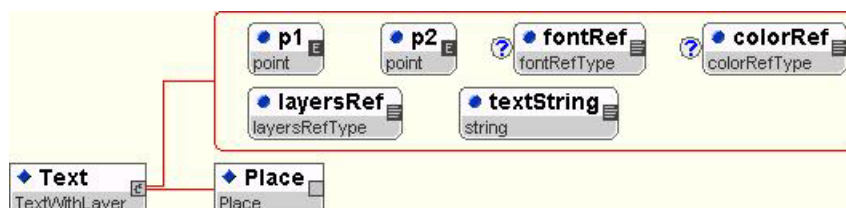
The attributes of the `Text` element are defined in 3.6.1 for the general case. The type for the `Text` element used in `Pattern` is TextWithLayers. This type adds the `layersRef` attribute to define the layers on which the `Text` is drawn.
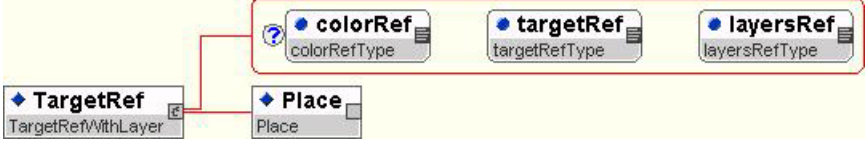
4.29.1.1   PadLocation

The `PadLocation` element references a `PadStack/@id`. The element adds an instance of the padstack to the pattern and places it relative to the origin of the pattern. The element also associates a pin name with the pad location. The attributes of a `PadLocation` element are defined as follows:

Attributes	Requirement	Description
`padStackRef`	REQUIRED	A reference to a `PadStack/@id` attribute.
`patternPinName`	REQUIRED	The pin name of this pattern that is associated with the land pattern pad location and the padstack referenced in this element.
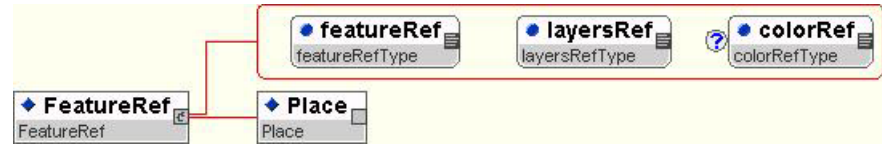
4.29.1.2   TargetRef

The `TargetRef` element references a `Target` to place an instance of the target in the pattern. A target is used for alignment or x-y registration (e.g. a fiducial), or as a bad-board indicator. The attributes of a `TargetRef` element are defined as follows:

Attributes	Requirement	Description
targetRef	REQUIRED	A reference to a `Target/@id`.
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers that the target referenced by `targetRef` is to be drawn.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.

4.29.1.3 FeatureRef

The `FeatureRef` element references a `Feature` to place an instance of the feature in the pattern relative to the origin of the pattern. The attributes of a `FeatureRef` element are defined as follows:

Attributes	Requirement	Description
featureRef	REQUIRED	A reference to a `Feature/@id`
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers that the feature referenced by `padRef` is to be drawn.
colorRef	OPTIONAL	A reference to a Color/@id attribute. The default color is black.

4.29.1.4 ArtWorkRef

The Artwork`Ref` element positions an artwork in a pattern. The `artworkRef` attribute references an `Artwork` element that is to be added to the `Pattern` on the layers referenced by `layersRef`. The attributes of an `ArtworkRef` element are defined as follows:

Attributes	Requirement	Description
artworkRef	REQUIRED	A reference to an `Artwork/@id` attribute.
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers that the `Artwork` referenced by `artworkRef` is to be drawn.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.

4.29.1.5   HoleRef

The `HoleRef` element positions a drilled or punched feature. The `HoleRef` element references `Hole/@id` attribute to add a hole to the `Pattern` The layers to be drilled or punched by a HoleRef are defined as part of the Hole. (By placing the layers reference in the `Hole` the process step, such as drilling hole for a buried via, are easier to identify.)

The `HoleRef` in patterns allows holes that aren't associated with a padstack to be added to a pattern. These holes would be used for mounting bolts or other non-electrical holes used in conjunction with a package. Do not use this `HoleRef` to create holes that are associated with pads or pins. The attributes of a `HoleRef` element are defined as follows:

Attributes	Requirement	Description
holeRef	REQUIRED	A reference to a `Hole/@id` attribute.
locationTolRef	OPTIONAL	A reference to a `TruePositionTolerance/@id` attribute. The tolerance defined by the referenced `TruePositionTolerance` applies to the tolerance allowed in the placement of the hole defined by the referenced `Hole`.

4.30   MountingLocations

The MountingLocations element defines a list of all the MountingLocation elements that are used by boards and panels.
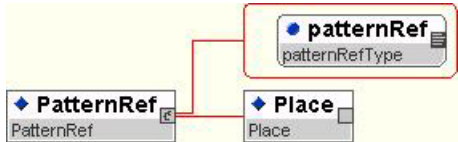
4.30.1   MountingLocation

An instance of a `MountingLocation` is placed on a board or a panel when it is referenced by a `MountingLocationRef` element. The attributes of a `MountingLocation` element are defined as follows:

Attributes	Requirement	Description
`refDes`	REQUIRED	The `refDes` attribute is a `qualifiedName` that uniquely identifies the `MountingLocation` within the GenCAM file. The `refDes` is referenced by a `MountingLocationRef/@refDes` attribute in order to create an instance of the `MountingLocation` on a board or a panel.
`layersRef`	REQUIRED	A reference to a `LayerSingle/@id` attribute. The `layersRef` defines the layer or layers of the board or panel on which the pattern and associated artwork is to be manufactured.

The attributes of the `Text` element are defined in 3.6.1 for the general case. The type for the `Text` element used in `MountingLocation` is TextWithLayers. This type adds the `layersRef` attribute to define the layers on which the `Text` is drawn.

104
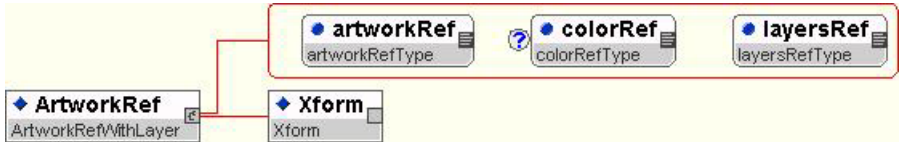
4.30.1.1   PatternRef

The `PatternRef` element references a `Pattern` element and places it relative to the origin of the `MountingLocation`. The parameters of a `PatternRef` element are defined as follows:

Attributes	Requirement	Description
patternRef	REQUIRED	A reference to a `Pattern/@id` attribute

4.30.1.2   ArtworkRef

The Artwork`Ref` element positions an artwork in the `MountingLocation`. The attributes of an `ArtworkRef` element are defined as follows:

Attribut es	Requirement	Description
artworkRef	REQUIRED	A reference to an `Artwork/@id` attribute.
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers that the `Artwork` referenced by `artworkRef` is to be drawn.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.

4.30.1.3 KeepOut

The `KeepOut` element defines an area associated with the `MountingLocation` that are not accessible. The origins of the `KeepOut` coincide with the placement location of the `MountingLocation`. The attributes of a `KeepOut` are defined as follows:

Attributes	Requirement	Description
type	REQUIRED	The type of the keepout. One of COMPONENT | VIA | ROUTE | TESTPIN | TESTPROBE | BOARD.
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers to which the keepout applies.

4.30.1.4 PatternPadRef

A PatternPadRef defined in a MountingLocation overrides the original definition of a pad for a specific pin and layer in the pattern referenced by the mounting location. Multiple PatternPadRefs can be defined for a specific pin and layer in the pattern. The original pad is removed and the new pads are drawn additively. If a PatternPadRef is defined without a padRef the pad at the pin and layer location is remove. This can be used to remove a land pattern pad from a pattern when a pin on the device is not connected to a route.

Attributes	Requirement	Description
padRef	OPTIONAL	A reference to a `Pad/@id` attribute. The pad defined by `Pad` is instanced relative to the point of origin of the pin referenced by the `patternPinRef` attribute.
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers that the pad referenced by `padRef` is to be drawn.
locationTolRef	OPTIONAL	A reference to a `TruePositionTolerance/@id` attribute. The tolerance defined by the referenced `TruePositionTolerance` applies to the tolerance allowed in the placement of the shape defined by the referenced pad.
patternPinRef	REQUIRED	A reference to `MountingLocation/PadLocation/@patternPinName` of the pattern referenced by `MountingLocation/PatternRef/@id`.

4.31   Packages

The `Packages` element defines a list of all the `Package` elements that are referenced in Device definitions. The physical dimensions of the package required to support manufacturing are described by a package definition.

4.31.1   Package

The `Package` element defines the physical dimensions of a device package. The `Package` definition includes information to support the placement and test of a device placed on a board or a panel. The location of a package on a panel or a board is determined when a ComponentPlacement references a device referencing the package. The attributes of a `Package` are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `Package` within the GenCAM file. The `id` is referenced by a `Device/@packageRef` attribute when a `Device` is placed on a board or a panel.

107

type	OPTIONAL	The package type is a general classification of physical shape and attachment. One of the following:
		CHIP | TANTALUM | MELF | EMBEDDED | SOT23 | SOT52 | OT89 | SOT143 | SOD123 | SOIC | SOPIC | SSOIC | TSOP | CERAMIC_FLATPACK | CERAMIC_QUAD_FLATPACK | PGA | PLASTIC_CHIP_CARRIER | LEADLESS_CERAMIC_CHIP_CARRIER | CERAMIC_DIP | PLASTIC_DIP | CERAMIC_SIP | PLASTIC_SIP | SQUARE_QUAD_FLATPACK | RECTANGULAR_QUAD_FLATPACK | SOJ | PLASTIC_BGA | CERAMIC_BGA | MINI_BGA | CHIP_SCALE | BARE_DIE | FLIPCHIP | AXIAL_LEADED | RADIAL_LEADED | TO_TYPE | MOLDED | POWER_TRANSISTOR | RELAY_SM | RELAY_TH | TRIMPOT_TH | TRIMPOT_SM | TRANSFORMER | CONNECTOR_SM | CONNECTOR_TH | COIL | CHOKE_SWITCH_SM | SWITCH_TH | HERMETIC_HYBRID | MCM | NETWORK
		If an appropriate type is not available in this check the list of package types is maintained at http://webstd.ipc.org/registered/package_types.
height	OPTIONAL	The maximum height of the package as measured from the finished surface of the board or panel on which it is mounted.
standoff	OPTIONAL	The clearance height to the bottom of the package body as measured from the finished surface of the board or panel on which it is mounted.
centroid	OPTIONAL	The centroid of the package as used by manufacturing equipment during assembly, inspection, and test. The package centroid is measured from the origin of the package.The centroid of the part is located at center of the bounding rectangle surrounding the body of the part and the patternPinRef graphics defined in all of the Pins defined in the package.
locationTolRef	OPTIONAL	A reference to a `TruePositionTolerance/@id` attribute. Defines the positioning tolerances of the `Package`.
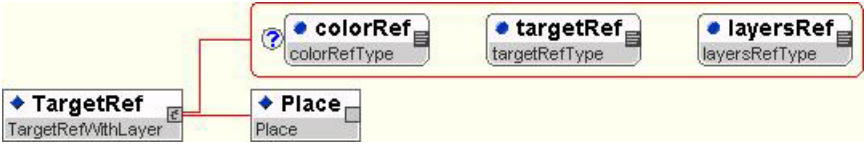
4.31.1.1 Body

The `Body` element defines the physical shape of the package outline. The body shape is a package outline that would be drawn on a display screen by a CAD or CAM tool or on a GenCAM drawing. Defining color-coded image that is to be displayed by a repair station is another possible use for a `Body` shape. The layers associated with the shape can be used to classify the function of a body artwork. The attributes of a `Body` element are defined as follows:

Attributes	Requirement	Description
artworkRef	REQUIRED	A reference to an `Artwork/@id` attribute. The shape defined by `Artwork` is instanced relative to the `Package` point of origin.
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers that the artwork referenced by artwork`Ref` is to be drawn.

4.31.1.2 TargetRef

The `TargetRef` element references a `Target` element. This places an instance of the target on the package. A target is used for alignment or x-y registration. The attributes of a `TargetRef` element are defined as follows:

Attributes	Requirement	Description
`targetRef`	REQUIRED	A reference to a `Target/@id`.
`layersRef`	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers that the target referenced by `targetRef` is located.
`colorRef`	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.

4.31.1.3 Pin
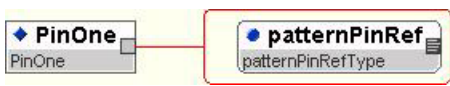
The `Pin` element defines the characteristics of each `Package` lead. The attributes of a `Pin` element are defined as follows:

Attributes	Requirement	Description
patternPinRef	REQUIRED	A reference to a Device/PinDesc/@patternPinRef. This attribute correlates the package physical pin name with the device physical pin name.
type	OPTIONAL	The pin type is one of the standard package pin types identified in JEDEC-1 (IEC-30). One of: TH_ROUND | TH_RIBBON | TH_V | TERMINATION | GULLWING | BUTTLEAD | JLEAD | SLEAD | WRAPAROUND | CASTELLATION | BALL | COLUMN | LAND
exit	OPTIONAL	The means by which a pin exits package. EDGE – the pin exits the package body at its edge. BOTTOM – the pin exits the package body from its bottom side. TOP – the pin exits the package body from its topside.
shadow	OPTIONAL	A reference to an `Artwork`/`@id` attribute that defines the two dimensional shape of the portion of the lead that extends beyond the package edge, looking down from the top of the package. This artwork is used to represent the shape of a part when displaying the part on a drawing or on the display screen of a CAD or CAM application.

4.31.1.4 PinOne

The `PinOne` element defines one of the pins in a package as the reference pin for the package. The only attribute of a `PinOne` element is defined as follows:

Attributes	Requirement	Description
patternPinRef	REQUIRED	A reference to a `Pin`/`@patternPinRef`. This attribute is a package physical pin name that is defined by the manufacturer to be pin 1 or the reference pin of the package.

4.32   Families

The Families element defines a list of all Family elements used within the GenCAM file.
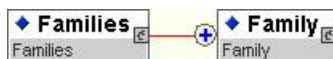
4.32.1   Family

The Family element defines the operational characteristics for pins on a device. The characteristics include the expected logical voltage characteristics of the family, the edge speed characteristics of the family, and the pin load characteristics of the family. The attributes of a Family are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the Family within the GenCAM file. The id is referenced by familyRef attributes.
driveHigh	REQUIRED	The minimum voltage level, which a device of the family will source from an output when the expected state is a logical 1.
driveLow	REQUIRED	The maximum voltage level that a device of the family will source from an output when the expected state is a logical 0.
receiveHigh	REQUIRED	The minimum voltage level, which must be driven for a device of the family to be expected to recognize a logical 1.
receiveLow	REQUIRED	The maximum voltage level for which a device of the family can be expected to recognize a logical 0.
openInputLogic	REQUIRED	The default logic-state exhibited by unconnected (floating) input pins. It is a fixed-field attribute of either ONE, ZERO, or X.
edgeSpeed	REQUIRED	The risetime in volts per nanosecond.
load	REQUIRED	The expected characteristics of the output pins by the surrounding circuitry is one of pulled up (UP), pulled down (DOWN), or left floating (NONE).

4.33   SchSymbols

The SchSymbols element defines a list of all SchSymbol elements used within the GenCAM file.
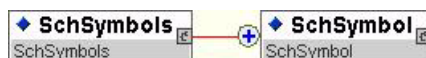
4.33.1   SchSymbol

Schematic or logic symbols are special user-defined patterns. They are used to describe the information for the characteristics of an electronic symbol intended to have a relationship to the package or the device containing the electronic elements. These symbols are most useful in attempting to take the same methodology used to draw the symbol in Drawings, and correlate it to the manner in which the logic elements are contained within a package. The attributes of a SchSymbol are defined as follows:
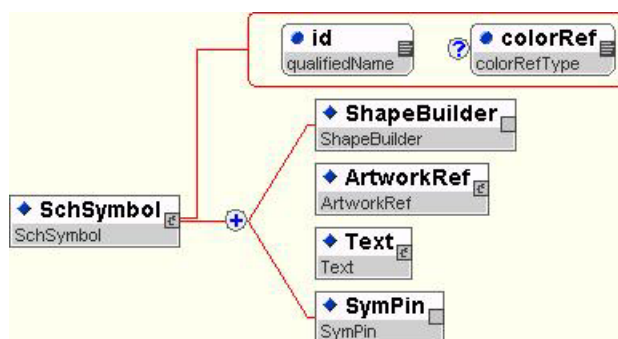
Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the SchSymbol within the GenCAM file. The id is referenced by schSymbolRef attributes.
colorRef	OPTIONAL	A reference to a Color/@id attribute. The default color is black.

4.33.1.1   ArtWorkRef

The Artwork_{Ref} element positions an artwork in the SchSymbol. The attributes of an ArtworkRef element are defined as follows:

Attributes	Requirement	Description
artworkRef	REQUIRED	A reference to an Artwork/@id attribute.
colorRef	OPTIONAL	A reference to a Color/@id attribute. The default color is black.

4.33.1.2   SymPin

The Sympin element defines a pin and the location of a pin on the symbol. The attributes of a Sympin element are defined as follows:

Attributes	Requirement	Description
pinGraphics	REQUIRED	A reference to an Artwork/@id attribute. The graphics on a symbol that are associated with a specific sympin on the symbol. The graphic is associated with a device pin when the symbol is placed on a drawing.
symPinName	REQUIRED	The pin name on the symbol. The pin name is associated with a pin on a device through the reference to Drawing/Sheet/Frame/SchSymbolRef/SymPinRef/@sympinRef

113

4.34 Devices

The `Devices` element defines a list of all `Device` elements used within the GenCAM file.
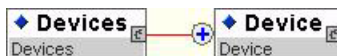
4.34.1 Device

A `Device` associates part numbers from an enterprise (a manufacturer or a distributor) with a set of device specific pin definitions and with a package definition. The geometry attributes of a device, including the location of the origin, are those of the package. The attributes of a `Device` are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the Device within the GenCAM file. The `id` is referenced by `deviceRef` attributes.
packageRef	REQUIRED	A reference to a `Package/@id` attribute.
familyRef	OPTIONAL	A reference to a `Family/@id`. This reference defines the logic family of the device. If an individual pin also references another logic family then the family referenced by a pin overrides the device level family.

114

type	REQUIRED	The type of the device. One of:
		RES – Used for two terminal resistors
		| VRES – Used for variable resistors
		| FABRES – Used for embedded resistors
		| RPCK – Used for resistor pack
		| DPCK – Used for diode pack
		| LEDPCK – Used for LED pack
		| HYBRID – Used for AtoD, DtoA
		| CAP – Used for two terminal non-polarized capacitors
		| VCAP – Used for variable capacitors
		| PCAP – Used for two terminal polarized capacitors
		| TCAP – Used for tantalum capacitors
		| FABCAP – Used for embedded capacitors
		| CPCK – Used for capacitor packs
		| IND – Used for inductors
		| VIND – Used for variable inductor
		| XFMR – Used for transformer
		| DIODE – Used for diodes, including Schottky
		| DIAC – Used for diacs
		| ZENER – Used for zener diodes
		| BRIDGE – Used for silicon bridge rectifier
		| PNP – Used for transistors, unijunctions and darlingtons.
		| NPN – Used for transistors, unijunctions and darlingtons.
		| NFET – Used for FET families
		| PFET – Used for FET families
		| NJFET – Used for FET families
		| PJFET – Used for FET families
		| TRIAC – Used for Triacs
		| SCR – Used for Thyristors
		| VR – Used for voltage regulators
		| OPTO – Used for opto-isolators
		| LED – Light Emitting Diode
		| OPAMP – Used for operational amplifier ICs
		| XTAL – Used for crystals
		| RELAY – Used for relays
		| SWITCH – Used for switches
		| FUSE – Used for fuses
		| JUMPER – Used for jumpers
		| CONN – Used for connectors
		| SOCKET – Used for sockets
		| LOGIC – Used for all logic devices
		| ANALOG – Used for analog ICs
		| OTHER – Used when none of the standard types are sufficient

115

4.34.1.1 Part

The `Part` element defines the enterprise supplying the part and the part identification information of the preferred part for this device. The attributes of a `Part` element are defined as follows:

Attributes	Requirement	Description
enterprisePart Id	REQUIRED	A reference to an Enterprise/@id attribute. The part identification number designated by the enterprise.
revision	OPTIONAL	The revision level of the part as designated by the enterprise.
enterpriseRef	REQUIRED	A reference to an `Enterprise/@id`.

4.34.1.2 Alias

The `Alias` element defines an alternative part number for a physically identical `Part`. An alias is used when the part is purchased from a different supplier, or if the supplier has alternative part numbers for the same part. (This might occur if the supplier uses different part numbers for different quantity purchases.) The attributes of an `Alias` element are defined as follows:

Attributes	Requirement	Description
enterpriseRef	REQUIRED	A reference to an Enterprise/@id attribute. The part identification number designated by the enterprise.
enterprisePart Id	REQUIRED	The revision level of the part as designated by the enterprise.
revision	OPTIONAL	A reference to an `Enterprise/@id`.

4.34.1.3   Alternate

The `Alternate` element defines an alternative part number for a functionally equivalent part that meets the manufacturing and design requirements. The alternate part may or may not be from a different enterprise. (The part may even be physically identical, but tested to different qualification specifications.) The attributes of an `Alternate` element are defined as follows:

Attributes	Requirement	Description
enterpriseRef	REQUIRED	A reference to an Enterprise/@id attribute. The part identification number designated by the enterprise.
enterprisePart Id	REQUIRED	The revision level of the part as designated by the enterprise.
revision	OPTIONAL	A reference to an `Enterprise/@id`.

4.34.1.4   PinDesc

The `PinDesc` element defines the functionality of a pin on the device. This element associates the input data for a tester and schematic symbol information with a pin. For example, the naming of the anode and cathode on diodes and zeners; collector, base and emitters on transistors; and gate, source, and drain on FETs are done using the PinDesc/@function attribute. The attributes of a `PinDesc` element are defined as follows:

Attributes	Requirement	Description
patternPinRef	REQUIRED	A reference to a Device/Package/Pin/@patternPinRef. This attribute correlates the package physical pin name with the device physical pin name.

117

function	OPTIONAL	One of the defined GenCAM pin functions.
		DRIVER – Used for outputs that drive a net.
		| RECEIVER – Used for inputs
		| BIDIRECTIONAL – Used for bi-directional logic
		| ANALOGIN – Used for analog ICs
		| ANALOGOUT – Used for analog ICs
		| NCLOSED – Used for relays and switches
		| NOPEN – Used for relays and switches
		| POWER – Used for power supply pins
		| GROUND – Used for ground pins
		| ANALOG – Used for analog pins
		| DIGITAL – Used for any family of logic pins
		| INACTIVE – Used for resistors, capacitors, etc. pins
		| ANODE – Used for diodes, zeners, unijunctions, thyristors etc.
		| CATHODE – Used for diodes, zeners, unijunctions, thyristors etc.
		| BASE – Used for transistors
		| COLLECTOR – Used for transistors (including IGBTs)
		| EMITTER – Used for transistors (including IGBTs)
		| SOURCE – Used for FETs
		| DRAIN | GATE – Used for FETs
		| WIPER – Used for variable components
		| CASE – Used for connection to device screen or can
		| CLOCK – Used for clock
		| ENABLE | DISABLE – Used for device enable/disable
		| TDI | TDO | TMS | TCK | TRST – Used for Boundary Scan
		| INTNC – Used for pins that are <u>internally</u> disconnected
		Additional values for the function attribute will be defined between releases of this document. They will be published at http://webstd.ipc.org/registered/PinDesc/@function.
circuitNumber	OPTIONAL	The circuit number that designates the circuit number in the IC package.
schSymbolRef	OPTIONAL	A reference to a `SchSymbol/@id` attribute.
symPinRef	OPTIONAL	A reference to the `SchSymbol/SymPin/@symPinName`. The `PinDesc` correlates other attribute references.
familyRef	OPTIONAL	Reference to a `Family/@id` This defines the logic family of the pin. If a family is assigned at the device level and to a specific pin then the local pin definition takes precedence over the device level assignment.

4.34.1.5   Program

The `Program` element defines the software content of the device. Use this `Program` element if the device is programmed prior to placement of the `Device` on the board, for example, devices that are programmed and placed back in inventory prior to assembly. If a device is to be programmed after assembly, the program definition is an element of ComponentPlacement. The attributes of a `Program` element are defined as follows:

Attributes	Requirement	Description
name	REQUIRED	The `name` attribute is a `string` that uniquely identifies the `Program` within the enterprise referenced by `enterpriseRef` file.
enterpriseRef	REQUIRED	A reference to an `Enterprise/@id` attribute. The enterprise that originated the program.
revision	REQUIRED	The program revision.
programDate	OPTIONAL	The program release date.

4.34.1.6   NumericValue

The `NumericValue` element defines a numeric value, units, and decade with minimum and maximum values that defines a measurement tolerance *window* that characterize a device. Minimum and maximum limits must be in the same units and decade as the nominal. When nominal, minimum and/or maximum attributes are present the minimum must be the least, maximum must be the greatest and the nominal must fall between these values. The attributes of a Numeric`Value` element are defined as follows:

Attributes	Requirement	Description
nominal	OPTIONAL	The measured value in the described units and decade
units	OPTIONAL	Units of measure
decade	OPTIONAL	Unit multiplier in powers of 10. (default is 0) Not applicable to non-numeric measures
minimum	OPTIONAL	Actual lower limit bound in the described units and decade
maximum	OPTIONAL	Actual upper limit bound in the described units and decade
position	OPTIONAL	Describe the positional location if the expression is for a multidimensional array of values.

comparator	OPTIONAL	One of EQ | NE | GT | LT | GE | LE | GTLT | GELE | GTLE | GELT | LTGT | LEGE | LTGE | LEGT
		EQ means equal to the nominal value. The nominal is a required attribute.
		NE means not equal to the nominal value. The nominal is a required attribute.
		GT means greater than the minimum. The minimum is a required attribute.
		LT means less than the maximum. The maximum is a required attribute.
		GE means greater than or equal to the minimum. The minimum is a required attribute.
		LE means less than or equal to the maximum. The maximum is a required attribute.
		GTLT means greater than the minimum and less than the maximum. Both limits are required attributes.
		GELE means greater than or equal to the minimum and less than or equal to the maximum. Both limits are required attributes.
		GTLE means greater than the minimum and less than or equal to the maximum. Both limits are required attributes.
		GELT means greater than or equal to the minimum and less than the maximum. Both limits are required attributes.
		LTGT means less than the minimum or greater than the maximum. Both limits are required attributes.
		LEGE means less than or equal to the minimum or greater than or equal to the maximum. Both limits are required attributes.
		LTGE means less than the minimum or greater than or equal to the upper limit. Both limits are required attributes.
		LEGT means less than or equal to the lower limit or greater than the upper limit. Both limits are required attributes.
		If no comparator is expressed the following applies:
		If both limits are present then the default must be GELE. (the nominal is optional).
		If only the upper limit is present then the default is LE.
		If only the lower limit is present then the default is GE.
		If only the nominal is present then the default is EQ.

4.34.1.7 *EnumeratedValue*

The `EnumeratedValue` element defines one of a fixed list of possible enumerated values that characterize a device. The type of value entered is always the same for all the same type of devices. For example, if the value used for a resistor is resistance, then all resistors have values that are resistance. The attributes of an `EnumeratedValue` element are defined as follows:

Attributes	Requirement	Description
`typeRef`	REQUIRED	A reference to a `registeredResource/@id` that defines the type of the value that is contained in the `value` attribute
`value`	REQUIRED	An enumerated value that matches one of the possible values defined by the `RegisteredResource` referenced by `typeRef`
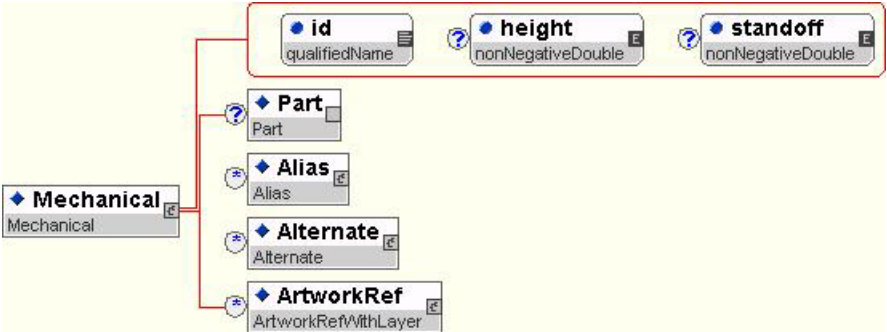
4.35   Mechanicals

The `Mechanicals` element defines a list of all `Mechanical` elements used within the GenCAM file.
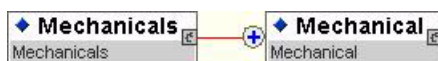
4.35.1   Mechanical
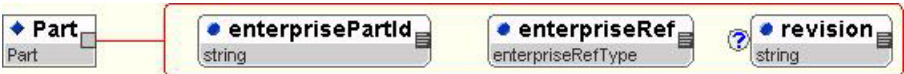
A `Mechanical` associates part numbers from an enterprise (a manufacturer or a distributor) with non-electrical features of a fixture, board, panel, or assembly. E.g. mechanical components, heat sinks, bolts, or alignment pins. A common example of these mechanical components would be mounting hardware for power transistors or card extractor mechanisms. The geometry attributes are defined relative to a local origin for the part definition. The attributes of a `Mechanical` are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the Mechanical within the GenCAM file. The id is referenced by mechanicalRef attributes.
height	OPTIONAL	The mechanical component's maximum height from the finished mounting surface.
standoff	OPTIONAL	The mechanical component's clearance from the finished mounting surface.

4.35.1.1 Part

The Part element defines the enterprise supplying the part and the part identification information of the preferred part for this device. The attributes of a Part element are defined as follows:

Attributes	Requirement	Description
enterprisePart Id	REQUIRED	A reference to an Enterprise/@id attribute. The part identification number designated by the enterprise.
revision	OPTIONAL	The revision level of the part as designated by the enterprise.
enterpriseRef	REQUIRED	A reference to an Enterprise/@id.

4.35.1.2 Alias

The Alias element defines an alternative part number for a physically identical Part. An alias is used when the part is purchased from a different supplier, or if the supplier has alternative part numbers for the same part. (This might occur if the supplier uses different part numbers for different quantity purchases.) The attributes of an Alias element are defined as follows:

Attributes	Requirement	Description
enterpriseRef	REQUIRED	A reference to an Enterprise/@id attribute. The part identification number designated by the enterprise.
enterprisePart Id	REQUIRED	The revision level of the part as designated by the enterprise.
revision	OPTIONAL	A reference to an Enterprise/@id.

4.35.1.3 Alternate

The Alternate element defines an alternative part number for a functionally equivalent part that meets the manufacturing and design requirements. The alternate part may or may not be from a different enterprise. (The part may even be physically identical, but tested to different qualification specifications.) The attributes of an Alternate element are defined as follows:

Attributes	Requirement	Description
enterpriseRef	REQUIRED	A reference to an Enterprise/@id attribute. The part identification number designated by the enterprise.
enterprisePart Id	REQUIRED	The revision level of the part as designated by the enterprise.
revision	OPTIONAL	A reference to an Enterprise/@id.

4.35.1.4 ArtWorkRef

The ArtworkRef element positions an artwork used by a Mechanical. The artwork in a Mechanical is used for representing the mechanical part in a GenCAM drawing or for representing the mechanical part on a display screen of a CAD or CAM tool.The attributes of an ArtworkRef element are defined as follows:

Attributes	Requirement	Description
artworkRef	REQUIRED	A reference to an `Artwork/@id` attribute.
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers that the `Artwork` referenced by `artworkRef` is to be drawn.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.
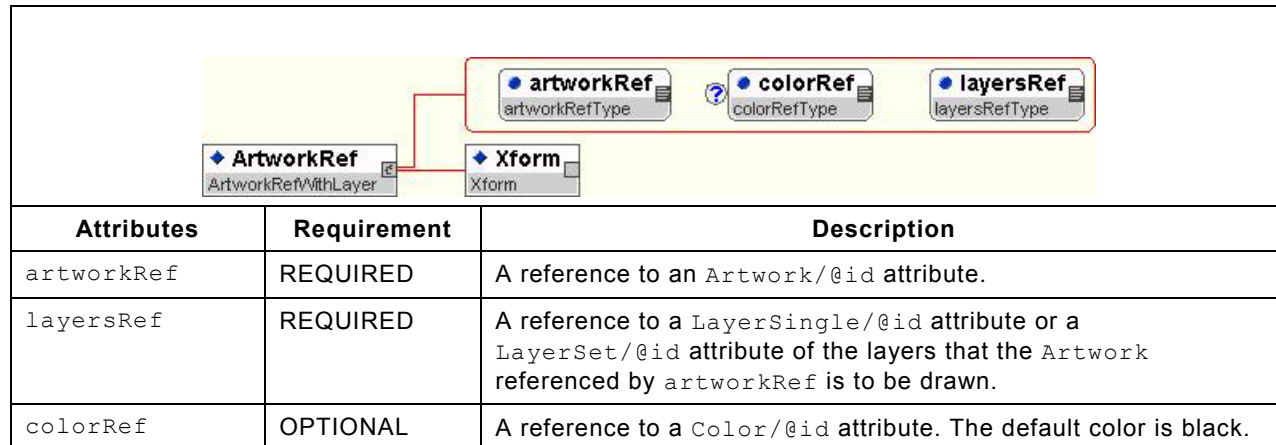
4.36   ComponentPlacements

The `ComponentPlacements` element defines a list of all `ComponentPlacement` elements used within the GenCAM file.
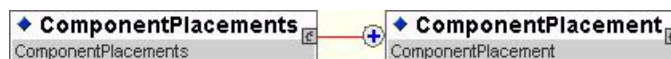
4.36.1   ComponentPlacement

A `ComponentPlacement` defines the location at which a device and associated mechanical parts are to be placed on a board or a panel. The `ComponentPlacement` is defined relative to the origin of a mounting location. The origins of the package in a device or the mechanical that are referenced by a `ComponentPlacement` coincide with the placement location of the `MountingLocation`. The attributes of a `ComponentPlacement` element are defined as follows:

Attributes	Requirement	Description
refDesRef	REQUIRED	The `refDesRef` attribute is a `qualifiedName` that uniquely identifies the `ComponentPlacement` within the GenCAM file. A `ComponentPlacementRef` /@refDesRef attribute will locate an instance of the device and mechanical parts defined in a `ComponentPlacement` at a MountingLocation/@refDes.
layersRef	REQUIRED	A reference to a `Layers`/@id attribute. The layer or layerset referenced defines the layers on the board on which the component is placed. If the `DeviceRef`/@process is FABRICATE then the device is fabricated on the layers. (Fabricated devices, such as thick film resistors, are manufactured as part of the board. They are not attached during the process of building a board or panel assembly.)

The images in Figure 40 illustrate the relationship between the orientation of a device placed on a board or panel and the the layer referenced by the `ComponentPlacment/@layerSingleRef` attribute and the `ComponentPlacment/Place/@mirror attribute`. If component is placed on a BOTTOM (as defined by the `surface` attribute) layer and one or more `LayerSwap` elements are defined in the layers section then for any layer referenced in the component definition, such as a layer inside a referenced pattern, for which a swap layer is defined will be swapped.

Note that four of the six boxes in Figure 40 are labeled with "abnormal state". The normal definition of a component is unlikely to use these parameter settings when placing parts on a board. These states are not illegal because under certain circumstances these states may be acceptable settings. Software that reads GenCAM data, such as the conformance test software, should flag the abnormal states as possible errors unless the abnormal state is recognized and accepted as correct. If an abnormal state is detected and flagged then it is up to the user to inspect the configuration of parts and patterns to determine if a real error has occurred.
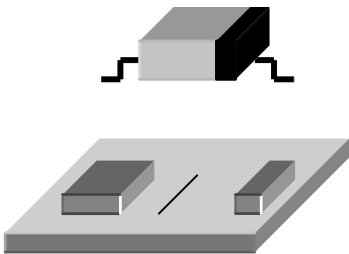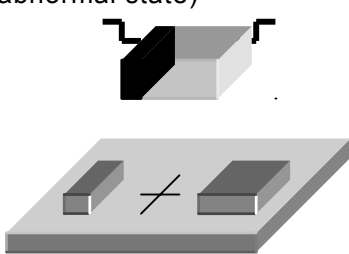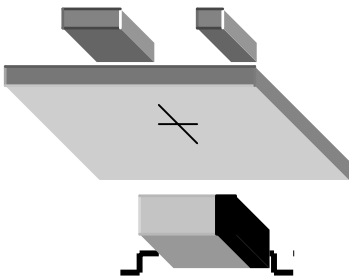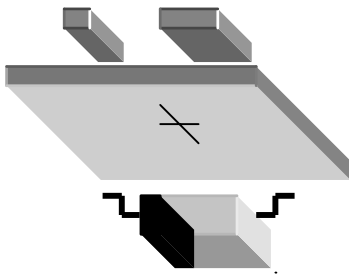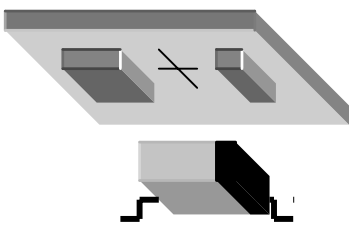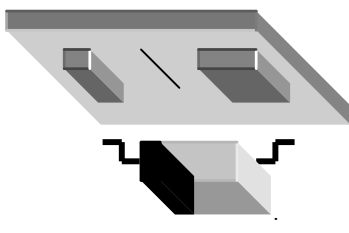
Effect of MountingLocation/Place/@mirror for this placement location, the surface selected by Mountinglocation/@layerSingleRef for this placement location, and Layers/LayerSwap on ComponentPlacement and all child elements			
Surface	Layer Swap	Not Mirrored	Mirrored
TOP	N/A		(abnormal state)
BOTTOM	No	(abnormal state)	(abnormal state)
BOTTOM	Yes	(abnormal state)	

Figure 40 Mirror and swap effects on components

4.36.1.1 *DeviceRef*
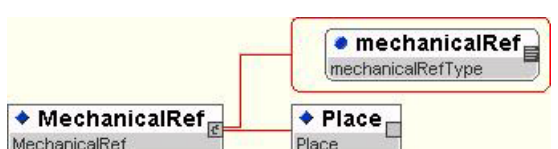
The DeviceRef element places a device at the location specified by the ComponentPlacement element. The element also indicates whether the device is manufactured as part of the board or panel or is assembled by placing a package containing the device on the board or panel. The attributes of a DeviceRef element are defined as follows:

126

Attributes	Requirement	Description
process	OPTIONAL	One of FABRICATE | PLACE. The process is set to FABRICATE if the part is manufactured when the board is manufactured. The Default is PLACE.
deviceRef	REQUIRED	A reference to a Device@/id

4.36.1.2   MechanicalRef

The MechanicalRef element places a mechanical part at the location specified by the ComponentPlacement element. The attributes of a MechanicalRef element are defined as follows:

Attributes	Requirement	Description
mechanicalRef	REQUIRED	References a Mechanical/@id

4.36.1.3   Program

The Program element defines the software content of the device at this location. Use this Program element if the device is programmed after placement of the Device on the board. The attributes of a Program element are defined as follows:

Attributes	Requirement	Description
name	REQUIRED	The name attribute is a string that uniquely identifies the Program within the enterprise referenced by enterpriseRef file.
enterpriseRef	REQUIRED	A reference to an Enterprise/@id attribute. The enterprise that originated the program.
revision	REQUIRED	The program revision.
programDate	OPTIONAL	The program release date.

127

4.36.1.4   PinDesc

This usage of the `PinDesc` element defines the functionality of a pin on the device at this `ComponentPlacement` location.   This element associates the input data for a tester and schematic symbol information with a pin. The pin function defines the actual function of the pin as it contributes to the mission of its associated component instance. For simple parts the pin fuction for the component may be identical to that of the pin function defined in the `Device` definition. However, for programmable devices many or all of the pin function attributes may be different for components of the same device type. For these pins the `PinDesc` will need to be defined at as part of `ComponentPlacement`. The attributes of a `PinDesc` element are defined as follows:

See the attribute values in the definition of PinDesc in section 4.34.1.4
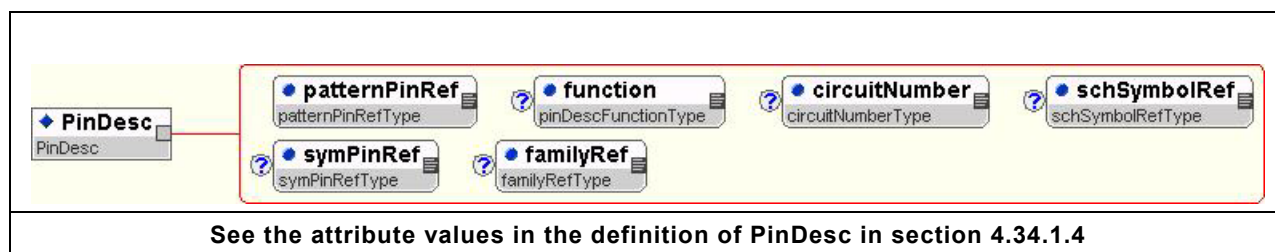
4.36.1.5   Preparation

The `Preparation` element defines the component lead conditions, both before and after insertion. The component lead condition is defined in a drawing referenced by `drawingRef`. The attributes of a `Preparation` element are defined as follows:

Attributes	Requirement	Description
name	REQUIRED	The name of the lead preparation.
type	REQUIRED	One of: LEADLENGTH, LEADBEND, CLINCH, CLINCHLENGTH, PREPCODE, or OTHER.
drawingRef	OPTIONAL	A reference to a `Drawing/@id`. The `Drawing` that is referenced contains a drawing of the preparation.

4.36.1.6   Presentation

The `Presentation` element defines the delivery system packaging media for the component. The attributes of a `Presentation` element are defined as follows:

Attributes	Requirement	Description
`deliverySystem`	REQUIRED	The part delivery presentation is one of REEL, PACK, BULK, TRAY, TUBE, or CARTRIDGE.
`configuration`	OPTIONAL	The characteristics of the media used.
`other`	OPTIONAL	Provides additional information regarding presentation.

4.36.1.7   KeepOut

The KeepOut element defines an area associated with the component that are not accessible to test or insertion machines. The origins of the `KeepOut` coincide with the placement location of the `ComponentPlacement`. The attributes of a `KeepOut` are defined as follows:

Attributes	Requirement	Description
`type`	REQUIRED	The type of the `KeepOut`. One of COMPONENT | VIA | ROUTE | TESTPIN | TESTPROBE | BOARD.
`layersRef`	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers to which the `KeepOut` applies.

4.37   Routes

The `Routes` element defines a list of all `Route` elements used within the GenCAM file.

129
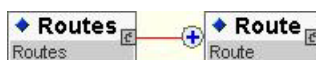
4.37.1   Route

The `Route` element defines the complete configuration of a net. This includes the circuit connectivity definition and the physical geometry of the traces and planes of the net. All geometry information defined in the route is defined relative to the origin of the route. The route origin coincides with the origin of the board or panel that is referencing the `Route` element. All shape elements of a net must be physically connected *(no opens)*. No part of a route geometry is allowed to touch the geometry of another route, or any other conducting geometry drawn on a board or a panel *(no shorts)*. For example, text drawn on a conductive layer would be a conducting geometry. The attributes of a `Route` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `Route` within the GenCAM file.

netClass	REQUIRED	The type of signal or function of this net. Some types of nets may require special consideration at in-circuit test.
		SIGNAL – a standard net
		CLK – a fast edge-speed net.
		FIXED – non-driven nets with fixed voltage levels that are derived from the circuit.
		GROUND – a ground net of the board.
		POWER – a power injection net for the board (I.e., connected to external power supply).
		UNUSED – the type to use when labelling route elements that are not intended to be connected to each other or as part of an active circuit.

4.37.1.1 Path

The Path element defines a conductor *(track)* width, width tolerance limit, and layout of a trace. The Path is part of a Route *(net)* definition. The polyline that defines the path *(trace)* geometry specifies the centerline of the trace. The attributes of a Path element are defined as follows:

Attributes	Requirement	Description
layerSingleRef	REQUIRED	A reference to a LayerSingle/@id attribute of the layers on which the polyline defined by the PolylineBuilder is to be drawn.
lineDescRef	REQUIRED	A reference to a lineDesc/@id attribute. characteristics of the line, such as the width, are defined by the LineDesc.
profileTolRef	OPTIONAL	A reference to a ProfileTolerance/@id attribute. This defines the tolerances on the trace.

131

4.37.1.2 Plane

The `Plane` element defines a polygon that is part of a `Route`. The attributes of a `Plane` are defined as follows:

Attributes	Requirement	Description
layersSingleRef	REQUIRED	A reference to a `LayerSingle/@id` attribute of the layer on which the polygon defined by the `PolygonBuilder` is to be drawn.
paintDescRef	REQUIRED	A reference to a `PaintDesc/@id` attribute. A void can be made in a plane by placing a second plane definition in the same route with the PaintDesc/@type set to VOID. The second plane defines a polygon in the shape of the voided area. They must use the same layersRef.

4.37.1.3 Via

The `Via` element defines a via in a `Route`. The attributes of a `Via` are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `Via` within the GenCAM file.
padStackRef	REQUIRED	A reference to a `PadStack/@id` attribute.
testPoint	OPTIONAL	The test probe placement offset. The offset is relative to the origin of the `Via`. The default places the test probe at the origin of the `Via`.

accessDesc	DEFAULT	The accessibility of a Via for test probing. One of the following values:
		ACCESS — can probe at this location from either top or bottom.
		ACCESSTOP — can probe at this location from top.
		ACCESSBOTTOM — can probe at this location from bottom.
		NOACCESS — can't get to this location.
		NOPROBE — don't probe here.
		PREFERRED — please probe here.
		MANDATORY — must probe here.
		The default value is NOACCESS.
		To be accessible by the test probe the Via must be on the top or bottom surfaces and should not be covered by the solder mask.

4.37.1.4   TestPad

The `TestPad` element defines a test pad in a `Route`. The attributes of a `TestPad` are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `TestPad` within the GenCAM file.
padStackRef	REQUIRED	A reference to a `PadStack/@id` attribute.
testPoint	OPTIONAL	The test probe placement offset. The offset is relative to the origin of the `TestPad`. The default places the test probe at the origin of the `TestPad`.

133

accessDesc	DEFAULT	The accessibility of a `TestPad` for test probing. One of the following values:
		ACCESS – can probe at this location from either top or bottom.
		ACCESSTOP – can probe at this location from top.
		ACCESSBOTTOM – can probe at this location from bottom.
		NOACCESS – can't get to this location.
		NOPROBE – don't probe here.
		PREFERRED – please probe here.
		MANDATORY – must probe here.
		The default value is ACCESS.
		To be accessible by the test probe the `TestPad` must be on the top or bottom surfaces and should not be covered by the solder mask.

4.37.1.5   ComponentPin

The `ComponentPin` element defines a component pin that is connected to the `Route`.   A `ComponentPin` element must be included for each component pin that is associated with the net. The attributes of a `ComponentPin` element are defined as follows:

Attributes	Requirement	Description
mountingLocation Ref	REQUIRED	A reference to a specific mounting location on the board or panel. The value of `mountingLocationRef` must be one of the unique key defined by: `MountingLocations/MountingLocation/@refDes`
patternPinRef	REQUIRED	A reference to a pin defined in the pattern at the mounting location. The pattern is defined by `PatternRef@patternRef` of the mounting location. The `patternPinRef` references the `Pattern/PadLocation/@patternPinName` of the mounting location referenced by `componentRef`.
testPoint	OPTIONAL	The test probe placement offset. The offset is relative to the origin of the pin referenced by `patternPinRef`. The default is to place the test probe at the origin of the referenced pin at the mounting location.

accessDesc	DEFAULT	The accessibility of a `ComponentPin` for test probing. One of the following values:
		ACCESS – can probe at this location from either top or bottom.
		ACCESSTOP – can probe at this location from top.
		ACCESSBOTTOM – can probe at this location from bottom.
		NOACCESS – can't get to this location.
		NOPROBE – don't probe here.
		PREFERRED – please probe here.
		MANDATORY – must probe here.
		The default value is ACCESS.
		To be accessible by the test probe the `ComponentPin` must be on the top or bottom surfaces and should not be covered by the solder mask.

4.37.2 HighPotTest

The `HighPotTest` element defines the testing attributes for a R$_{out}$e that must be tested using high voltage and current to test board performance capability. The attributes of a `HighPotTest` element are defined as follows:

Attributes	Requirement	Description
ohmValue	REQUIRED	The value in Ohms expected for this circuit.
posTol	OPTIONAL	The deviation above the `ohmValue` that is allowed.
negTol	OPTIONAL	The deviation below the `ohmValue` that is allowed.
testVoltage	REQUIRED	The maximum voltage to be used for the `HighPotTest`.
routeRef	REQUIRED	A reference to a `Route/@id`. This is the "ground" net for the test.

4.37.2.1 UnbalancedImpedance

The `UnbalancedImpedance` element defines the expected impedance to be measured in an unbalanced impedance test. The attributes of an `UnbalancedImpedance` element are defined as follows:

Attributes	Requirement	Description
ohmValue	REQUIRED	The value in Ohms expected for this circuit.
negTol	OPTIONAL	The deviation above the ohmValue that is allowed.
posTol	OPTIONAL	The deviation below the ohmValue that is allowed.
routePlaneRef	REQUIRED	A reference to a Route/@id. This is should be a ground plane for the net being tested.

4.37.2.2   *BalancedImpedance*

The BalancedImpedance element defines the expected impedance to be measured in a balanced impedance test.   The attributes of a BalancedImpedance element are defined as follows:

Attributes	Requirement	Description
ohmValue	REQUIRED	The value in Ohms expected for this circuit.
negTol	OPTIONAL	The deviation above the ohmValue that is allowed.
posTol	OPTIONAL	The deviation below the ohmValue that is allowed.
routeRef	REQUIRED	A reference to a Route/@id. This is the second other half of the net pair required for a balanced impedance test.
routePlaneRef	REQUIRED	A reference to a Route/@id. This is should be a ground plane for the net being tested.

4.37.2.3   Capacitance

The `Capacitance` element defines the expected impedance to be measured in a capacitance test. The attributes of a Capacitance element are defined as follows:

Attributes	Requirement	Description
faradValue	REQUIRED	The value in Farads expected for this circuit.
negTol	OPTIONAL	The deviation above the `faradValue` that is allowed.
posTol	OPTIONAL	The deviation below the `faradValue` that is allowed.
routeRef	REQUIRED	A reference to a `Route/@id`. This is the second other half of the net pair required for a capacitance test.

4.37.2.4   Inductance

The Inductance element defines the expected impedance to be measured in a Inductance test. The attributes of a Inductance element are defined as follows:
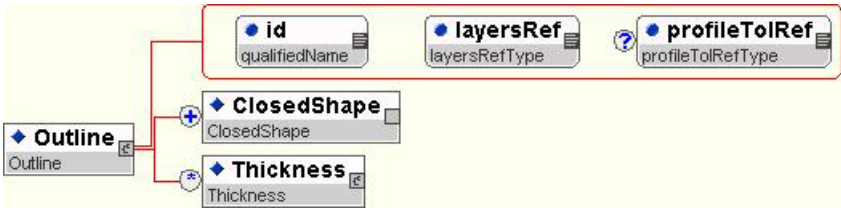
Attributes	Requirement	Description
henryValue	REQUIRED	The value in Henrys expected for this circuit.
negTol	OPTIONAL	The deviation above the `henryValue` that is allowed.
posTol	OPTIONAL	The deviation below the `henryValue` that is allowed.

4.38   Elements used by more than one product type

The elements in this section are used by more than one type of product. The definition of the elements will be defined one in this section and then those definitions will be referenced by the Product definitions.

4.38.1   Outline

The `Outline` element defines the outer periphery of this product. All `Outline` elements share a common point of origin with the point of origin of the product. The attributes of an `Outline` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the Outline within the GenCAM file.
layersRef	REQUIRED	A reference to a LayerSingle/@id attribute or a LayerSet/@id attribute of the layers to which the shape defined by ClosedShape applies.
profileTolRef	OPTIONAL	A reference to a ProfileTolerance/@id attribute. Defines the tolerances on the outer periphery dimensions.

4.38.2   Cutout

The Cutout element defines an internal area of a this product that is to have one or more layers cut away. The Cutout is defined relative to the origins of the product. The attributes of a Cutout element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the Cutout within the GenCAM file.
barrelDescRef	OPTIONAL	A reference to a BarrelDesc/@id attribute. The plating and coating attributes apply to both the walls and the bottom of a Cutout.
profileTolRef	OPTIONAL	A reference to a ProfileTolerance/@id attribute. Defines the tolerances on the Cutout dimensions.
layersRef	REQUIRED	A reference to a LayerSingle/@id attribute or a LayerSet/@id attribute of the layers to which the shape defined by ClosedShape applies.
locationTolRef	OPTIONAL	A reference to a TruePositionTolerance/@id attribute. Defines the positioning tolerances of the Cutout.

4.38.3   Well

The `Well` element defines an internal area of this product in which material of a specific depth is removed. The well is defined relative to the origins of the product. The attributes of a `Well` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `Cutout` within the GenCAM file.
nominalAngle	REQUIRED	The nominal angle between the sidewalls of the `Well`. An angle of zero defines vertical, parallel walls.
surface	REQUIRED	The side from which the `Well` is milled. One of TOP, BOTTOM or BOTH.
cutMode	REQUIRED	One of: REMOVE - the `remValue` specifies the material to be removed. REMAIN - the `remValue` specifies the remaining material.
remValue	REQUIRED	The depth of the material to be removed or the depth of the material that remains.
barrelDescRef	OPTIONAL	A reference to a `BarrelDesc/@id` attribute. The plating and coating attributes apply to both the walls and the bottom of a `Well`.
profileTolRef	OPTIONAL	A reference to a `ProfileTolerance/@id` attribute. The tolerances on the size of the finished `Well` after plating or coating.
locationTolRef	OPTIONAL	A reference to a `TruePositionTolerance/@id` attribute. Defines the positioning tolerances of the `Well`.

4.38.4 Slot

The Slot element defines a slot that is cut through one or more layers of the product. The slot is defined relative to the origins of the product. The attributes of a Slot element are defined as follows:

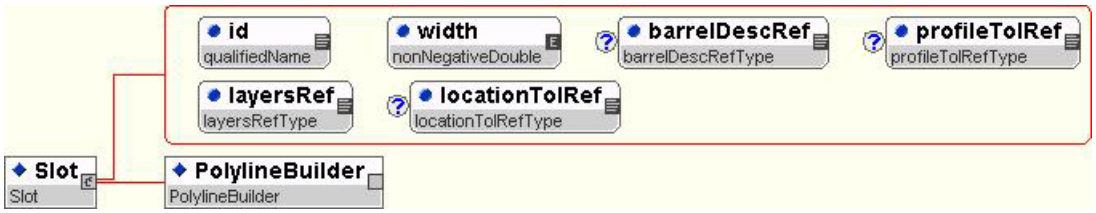| Attributes	Requirement	Description
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the Slot within the GenCAM file.
width	REQUIRED	The width of the slot. The PolylineBuilder of the Slot element defines the centerline along which the slot is cut. All corners of the slot and the ends extending past the ends of the polyline of the slot will have a radius equal to ½ the width. The polyline is defined relative to the origins of the product. This width overrides the lineDesc of the polyline.
barrelDescRef	OPTIONAL	A reference to a BarrelDesc/@id attribute. The plating and coating attributes apply to both the walls and the bottom of a Slot.
profileTolRef	OPTIONAL	A reference to a ProfileTolerance/@id attribute. The tolerances on the size of the finished Slot after plating or coating.
layersRef	OPTIONAL	A reference to a LayerSingle/@id attribute or a LayerSet/@id attribute of the layers to which the shape defined by PolylineBuilder applies.
locationTolRef	OPTIONAL	A reference to a TruePositionTolerance/@id attribute. Defines the positioning tolerances of the Slot.

4.38.5   Keepout

The `KeepOut` element defines an inaccessible area of the product. The `KeepOut` is defined relative to the origins of the product. The attributes of a `KeepOut` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `KeepOut` within the GenCAM file.
type	REQUIRED	One of COMPONENT, VIA, ROUTE, TESTPIN, TESTPROBE, or BOARD.
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers to which the shape defined by `ClosedShape` applies.

4.38.6   HoleRef

The `HoleRef` element positions a drilled or punched feature. The `HoleRef` element references `Hole/@id` attribute to add an instance of the hole to the parent element. The layers to be drilled or punched by a `HoleRef` are defined by `Hole`. (By placing the layers reference in the Hole the process step, such as drilling hole for a buried via, are easier to identify.) The attributes of a `HoleRef` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `HoleRef` within the GenCAM file.
holeRef	REQUIRED	A reference to a `Hole/@id` attribute. Holes located in this product element are for mounting, alignment pins, or other non-electrical uses. Holes used to attach devices should be defined in a `PadStack` or a `Pattern`.
locationTolRef	OPTIONAL	A reference to a `TruePositionTolerance/@id` attribute. The tolerance defined by the referenced `TruePositionTolerance` applies to the tolerance allowed in the placement of the hole defined by the referenced `Hole`.

4.38.7   Groove

The `Groove` element defines a groove cut in the board for segmentation or other purposes. The attributes of a `Groove` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `Groove` within the GenCAM file.
nominalAngle	REQUIRED	the nominal angle between the sidewalls of the `Groove`. An angle of zero defines vertical, parallel walls.
surface	REQUIRED	The side on which the groove is to be ground TOP, BOTTOM or BOTH.
cutMode	REQUIRED	One of: REMOVE - the `remValue` specifies the material to be removed. REMAIN - the `remValue` specifies the remaining material.
remValue	REQUIRED	The depth of the material to be removed or the depth of the material that remains.
grooveWidth	OPTIONAL	The width of the `Groove` at the surface of the panel. The `PolylineBuilder` of the `Groove` element defines the centerline along which the groove is cut. All corners of the groove and the ends extending past the ends of the polyline of the groove will have a radius equal to ½ the `grooveWidth`. The polyline is defined relative to the origins of the product.
profileTolRef	OPTIONAL	A reference to a `ProfileTolerance/@id` attribute. The tolerances on the size of the finished `Groove` after plating or coating.
locationTolRef	OPTIONAL	A reference to a `TruePositionTolerance/@id` attribute. The tolerance defined by the referenced `TruePositionTolerance` applies to the tolerance allowed in the placement of the groove.

4.38.8   TargetRef

The `TargetRef` element references a `Target` to place an instance of the target on the product. A target is used for alignment or x-y registration (e.g. a fiducial), or as a bad-board indicator. The attributes of a `TargetRef` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `TargetRef` within the GenCAM file.
targetRef	REQUIRED	A reference to a `Target/@id`.
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers that the target referenced by `targetRef` is to be drawn.
colorRef	OPTIONAL	A reference to a `Color/@id` attribute. The default color is black.

4.38.9   LogoRef

The `LogoRef` element places an instance of the logo in the product relative to the origin of the product. The attributes of a `LogoRef` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `LogoRef` within the GenCAM file.
logoRef	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `LogoRef` within the GenCAM file.
colorRef	OPTIONAL	A reference to a Color/@id attribute. The default color is black.
layersRef	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers to which the shape defined by `Logo` applies.

4.38.10 Images

The `Images` element defines a list of all `Images` elements used within the GenCAM file.
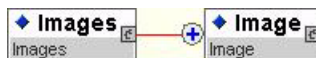
4.38.10.1 Image

The `Image` element draws images on a board or panel. If an image is drawn on the board the consistency level must be set to one of IMAGE_ONLY, IMAGE_NO_MATCH, IMAGE_MATCHES, or IMAGE_NO_NETS. Images can potentially touch and short routes. The consistency flags should reflect when this is a possibility. If an `Image` is drawn on a conducting layer and is part of a net the Route associated with an `Image` element is referenced through the `Image/@source` attribute. The attributes of an `Image` element are defined as follows:

Attributes	Requirement	Description
`id`	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `Image` within the GenCAM file.
`imageRef`	REQUIRED	A reference to the `id` attribute of one of: `Artwork`, `Feature`, `Pattern`, or `SchSymbol` The `Pattern` reference will ignore the holes in the pattern and the padstacks.
`layersRef`	REQUIRED	A reference to a `LayerSingle/@id` attribute or a `LayerSet/@id` attribute of the layers on which the image referenced by `imageRef` is to be drawn.

consistencyLevel	REQUIRED	The consistencyLevel indicates whether the `Image` data in the Board section is derived directly from, and is geometrically identical, to the geometry and netlist data that is represented in ROUTES, PATTERNS, and PADSTACKS sections of the file. The value of consistencyLevel is defined as one of:
		
		IMAGEONLY
		
		    then the BOARD contains no drawing information for the layers other than as IMAGE elements.
		
		IMAGENOMATCH
		
		    the geometry information drawn may not match the geometry information that is contained in ROUTES, PATTERNS, and PADSTACKS. Routes could be shorted.
		
		IMAGEMATCHES
		
		    then the IMAGE data duplicates the layer geometry information that is contained in ROUTES, PATTERNS, and PADSTACKS.
		
		IMAGENONETS
		
		    the images on the board do not touch any of the routes.

colorRef	OPTIONAL	A reference to a Color/@id attribute. The default color is black.
source	OPTIONAL	An xpath referencing the shape element corresponding to the `Image` in a `Route` or `MountingLocation`. The source is required if `consistencyLevel` is IMAGEMATCHES.

4.38.11 MountingLocationRefs

The `MountingLocationRefs` element defines a list of all `MountingLocationRef` elements used within the GenCAM file.
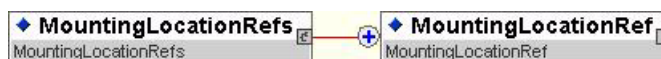
4.38.11.1 MountingLocationRef

An instance of a `MountingLocation` is placed on a board or a panel when it is referenced by a `MountingLocationRef` element. The attributes of a `MountingLocation` element are defined as follows:

Attributes	Requirement	Description
refDesRef	REQUIRED	A reference to a `MountingLocation/@refDes attribute`.
profileTolRef	OPTIONAL	A reference to a `ProfileTolerance/@id` attribute. The `ProfileTolerance` defines the tolerance on the placement of the `MountingLocation`.

145

4.38.12 RouteRefs

The `RouteRefs` element defines a list of all `RouteRef` elements used within the GenCAM file.
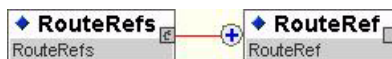
4.38.12.1 RouteRef

An instance of a `Route` is placed on a board or a panel when it is referenced by a `RouteRef` element. The attributes of a `RouteRef` element are defined as follows:

Attributes	Requirement	Description
routeRef	REQUIRED	A reference to a `Route/@id attribute`.

4.38.13 DesignRules

The `DesignRules` element defines a list of all `DesignRuleSet` elements used by the product.
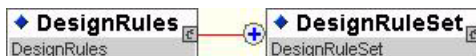
4.38.13.1 DesignRuleSet

A DesignRuleSet element references an externally defined set of design rules that are to be applied to this product. These design rules may be defined in IPC performance standards, such as IPC-6012. The XML Schema that is defined in the associated IPC performance standard determines the content and format of the referenced XML document.

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `DesignRule` within the GenCAM file.
comment	OPTIONAL	A string that describes any unique characteristics or instructions for this design rule.
ruleSetURL	REQUIRED	A URI that references a collection of design rules that apply to this product. The content and format of the design rules are defined by IPC performance standards.

4.38.13.2 ResultSet

A `ResultSet` element references the results of design rule test verification. This is an externally defined set of design rules that are to be applied to this product. These design rules may be defined in IPC performance standards, such as IPC-6012 The XML Schema that is defined in the associated IPC performance standard determines the content and format of the referenced XML document. The design rules may also be manufacturer specific.

Attributes	Requirement	Description
comment	REQUIRED	A string that describes the results of an audit of according to the specified design rules.
resultSetURL	OPTIONAL	A URI that references a collection of design rules that apply to this product. The content and format of the design rules are defined by IPC performance standards.
fileRevisionRef	REQUIRED	The fileRevisionRef can be a simple as the addition of comments to the design rules or as complex as a completely new GenCAM file that addresses all the variations needed to make the product(s) manufacturable.

4.38.14 BareBoardTest

The `BareBoardTest` element defines the test voltage and current to be used for board continuity testing. The attributes of a `BareBoardTest` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `BareBoardTest` within the GenCAM file.
continuityVoltage	REQUIRED	The continuity voltage in volts.
continuityCurrent	REQUIRED	The continuity current in amps.
isolationVoltage	REQUIRED	The isolation voltage in volts.

4.38.15 Placement

The `Placement` statement places a product on a product. The product is placed relative to the origin of the referencing product.

Attributes	Requireme nt	Description
`productRef`	REQUIRED	When used with by a `Panel` the `Placement` statement references the `id` attribute of a `Panel` product or a `Board` product.
		When used with by in `Fixture` product the `Placement` statement references the `id` attribute of a `Board` product, an `Assembly` product, or a `Board` product.
`productInstanceId`	REQUIRED	The identifier for the instance of a product on a panel
		The productInstanceIdType must match "[a-zA-Z][a-zA-Z0-9_-]*".
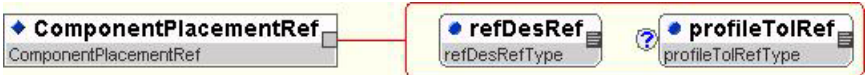
4.38.16 ComponentPlacementRefs

The `ComponentPlacementRefs` element defines a list of all `ComponentPlacementRef` elements used within the GenCAM file.

4.38.16.1 ComponentPlacementRef

A `ComponentPlacementRef` places a device and associated mechanical parts on a board or a panel. The origins of the package in a device or the mechanical part coincide with the placement location of the `MountingLocation`. The attributes of a `ComponentPlacementRef` element are defined as follows:

Attributes	Requirement	Description
`refDesRef`	REQUIRED	A reference to a `MountingLocation/@refDes attribute`.
`profileTolRef`	OPTIONAL	A reference to a `ProfileTolerance/@id` attribute. The `ProfileTolerance` defines the tolerance on the placement of the device and mechanical parts referenced contained in the referenced `ComponentPlacement`.

4.38.17 PowerSupplyRefs

The `PowerSupplyRefs` element defines a list of all `PowerSupplyRef` elements used within the GenCAM file.

4.38.17.1 PowerSupplyRef

The `PowerSupplyRef` references a `PowerSupply` definition that is to be used by the `Assembly`, `Panel` or `Fixture`.

Attributes	Requirement	Description
powerSupplyRef	REQUIRED	A reference to a `PowerSupply/@id` attribute.

4.38.18 TestConnectRefs

The `TestConnectRefs` element defines a list of all `TestConnectRef` elements used within the GenCAM file.
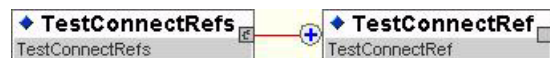
4.38.18.1 TestConnectRef

The `TestConnectRef` references a `TestConnect` definition that is to be used by the `Assembly or Panel`.

Attributes	Requirement	Description
testConnectRef	REQUIRED	A reference to a `TestConnect/@id` attribute.

4.39   Boards
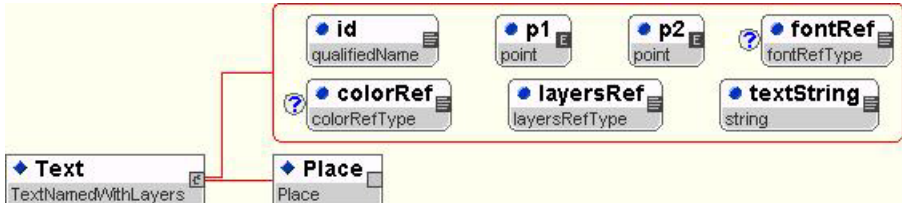
The `Boards` element defines a list of all `Board` elements used within the GenCAM file.
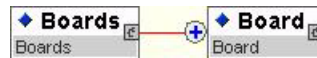
4.39.1   Board

The `Board` element defines a printed board or printed board coupon. A board is one of four types of `Product` that can be defined in a GenCAM file. The only attribute of a `Board` element is defined as follows:

Attributes	Requirement	Description
boardRef	REQUIRED	A reference to a `Product/@id` where the `Product/@type` is Board.

The elements defined in section 4.38 include: `Outline`, `Cutout`, `Well`, `Slot`, `KeepOut`, `HoleRef`, `Grove`, `TargetRef`, `LogoRef`, `Images`, `MountingLocationRefs`, `RouteRefs`, and `BareBoardTest`

The Text element is defined in 3.6.1 for the general case. The type for Text used on a Board is TextNamedWithLayers. This type adds additional attributes. The id is added because the Text is defined as a top-level element and as such needs to have an id. The layersRef attribute defines which the layers on which the Text is drawn.

4.40   PowerSupplies

The `Power` section defines power supplies used to power the board or panel during test. The following XML defines the elements allowed in the `Power` section, and the constraints on their use:
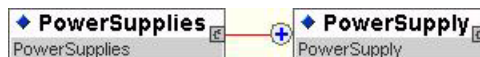
4.41   PowerSupply

The `PowerSupply` element defines test system power supply attributes, and associates the attributes with a net for the power signal and the reference return path net. The attributes of a `PowerSupply` element are defined as follows:

Attributes	Requirement	Description
`id`	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `PowerSupply` within the GenCAM file.
`currentLimit`	REQUIRED	The maximum current limit in amps.
`routeRef`	REQUIRED	A reference to a `Route/@id attribute`. The net connected to the power signal.
`routePlaneRef`	REQUIRED	A reference to a `Route/@id attribute`. This is the reference net for the ground plane.

4.42   Assemblies

The `Assemblies` element defines a list of all `Assembly` elements used within the GenCAM file.
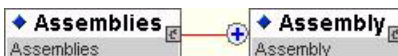
4.42.1   Assembly

The `Assembly` element defines all the items included in an `Assembly`. The attributes of an `Assembly` element are defined as follows:

Attributes	Requirement	Description
assemblyRef	REQUIRED	A reference to a Product/@id where Product/@type is ASSEMBLY.
boardOrPanelRef	REQUIRED	A reference to a Product/@id where Product/@type is a Board or a Panel.
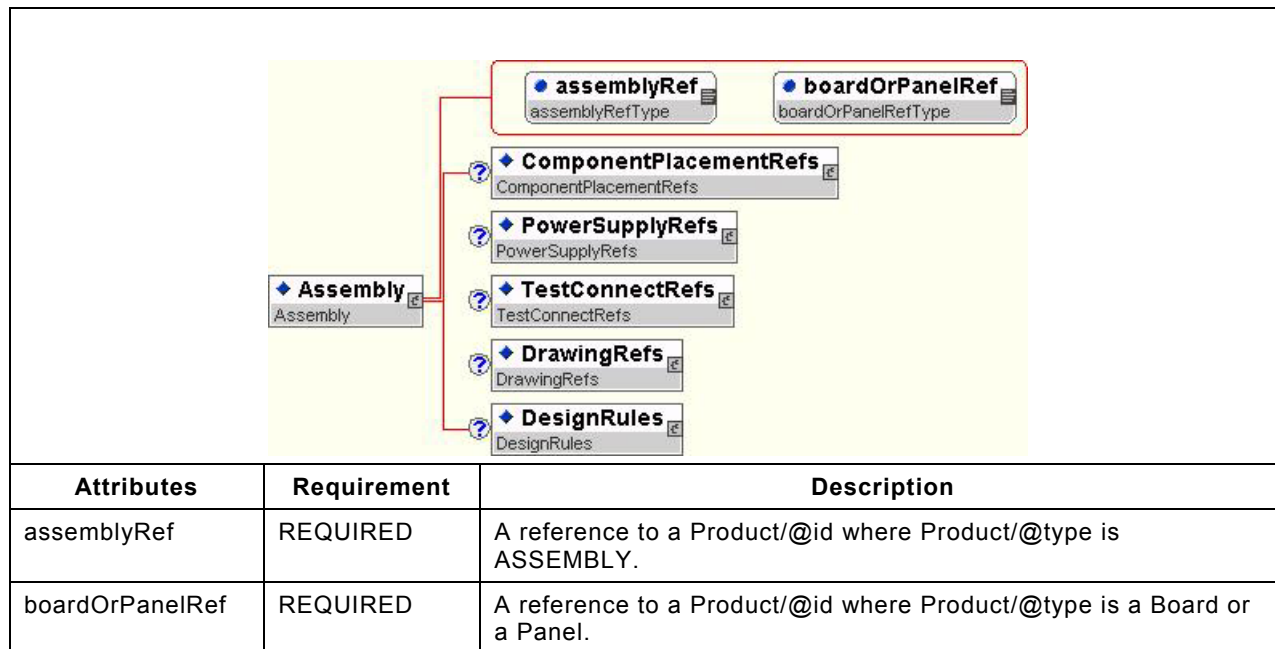
4.42.1.1   *DrawingRefs*

The DrawingRefs element defines a list of all DrawingRef elements used within the GenCAM file.
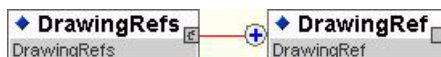
4.42.1.1.1   *DrawingRef*

The `DrawingRef` references drawings of the assembly. The attributes of an `DrawingRef` element are defined as follows:

Attributes	Requirement	Description
drawingRef	REQUIRED	A reference to a `Drawing/@id attribute.`
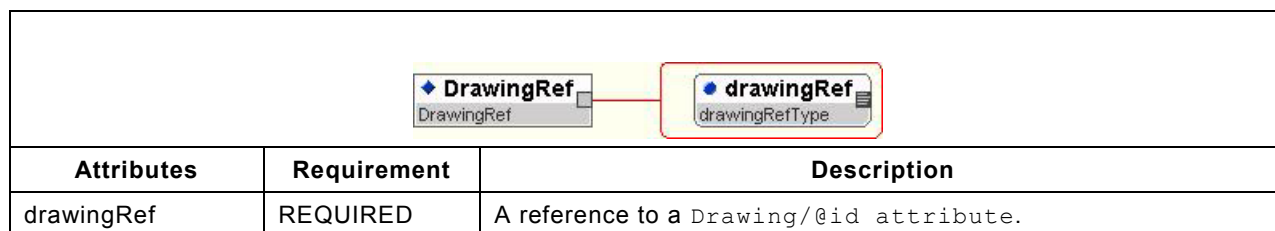
4.43   Panels
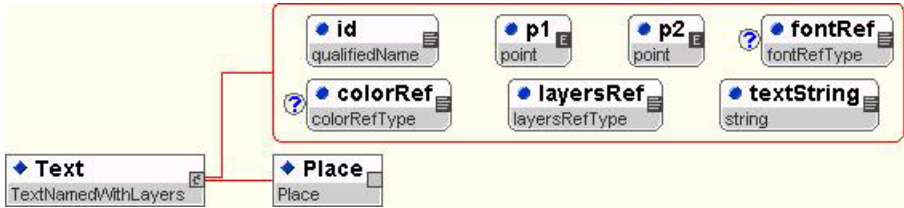
The `Panels` element defines a list of all `Panel` elements used within the GenCAM file.
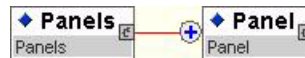
4.43.1   Panel

The `Panel` element defines a printed board panel or subpanel. The definitions of panels include features such as the outer shape of the panel, instances of boards, and instances of subpanels. The only attribute of a `Panel` element is defined as follows:

Attributes	Requirement	Description
`panelRef`	REQUIRED	A reference to a Product/@id where Product/@type is PANEL.
Elements defined in section 4.38:`Outline`, `Cutout`, `Well`, `Slot`, `KeepOut`, `HoleRef`, `Grove`, `TargetRef`, `LogoRef`, `Images`, `MountingLocationRefs`, `RouteRefs`, and `BareBoardTest`		

The Text element is defined in 3.6.1 for the general case. The type for Text used on a Panel is TextNamedWithLayers. This type adds additional attributes. The id is added because the Text is defined as a top-level element and as such needs to have an id. The layersRef attribute defines which the layers on which the Text is drawn.

4.44   Fixtures

The `Fixtures` element defines a list of all `Fixture` elements used within the GenCAM file.

155

4.44.1   Fixture

A `Fixture` element defines a fixture that holds a board, panel, or assembly during the manufacturing or test process. Typical applications that use fixtures include bare-board test, in-circuit test, and the application of glue-dots. The attributes of a `Fixture` element are defined as follows:

Attributes	Requirement	Description
fixtureRef	REQUIRED	A reference to a Product/@id where Product/@type is FIXTURE.
application	REQUIRED	The primary fixture application. One of BDFAB – IPC-2514 (board fabrication) BDTST – IPC-2515 (board test) BDASM – IPC-2516 (board assembly) ASEMT – IPC-2517 (assembly test)
function	OPTIONAL	ICT – (in-circuit test) BAREBOARDTEST SOLDERSTENCIL GLUEDOT ASSEMBLY OTHER – use a name that hints at the fiture type.

4.45   Drawings

The `Drawings` element defines a list of all `Drawing` elements used within the GenCAM file.

4.45.1   Drawing

A `Drawing` defines a set of drawings associated with a product (board, panel, assembly, or fixture). The `Drawing` element is used to identify a single sheet or multiple sheet drawing. The attributes of a `Drawing` element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `Drawing` within the GenCAM file.
number	REQUIRED	the drawing number, associated with the drawing.
revision	REQUIRED	the revision level of the drawing.
type	REQUIRED	the type of the drawing as one of FABRICATION, ASSEMBLY, SCHEMATIC, DETAIL, SPECIFICATION BOARDTEST, ASSEMBLYTEST, MACHINE, PHOTOTOOL, COMBINATION or PARTSLIST.
date	REQUIRED	the date code in the standard GenCAM date format.
size	OPTIONAL	the paper size of the drawing as one of A, B, C, D, E, A0, A1, A2, A3 or A4.
name	REQUIRED	A descriptive name or title for the drawing.

4.45.1.1   Sheet

The `Sheet` element defines drawing sheets in the drawing set. The attributes of a `Sheet` element are defined as follows:

Attributes	Requirement	Description
number	REQUIRED	The sheet number of a single or a multi-sheet drawing.
revision	REQUIRED	The revision number for the sheet.

4.45.1.1.1   Frame

The `Frame` element defines a graphic border template for a drawing sheet. The attributes of a `Frame` element are defined as follows:

Attributes	Requirement	Description
artworkRef	REQUIRED	Areference to an Artwork/@id. The artwork contains the outline for the drawing border, zone identifiers, and the drawing title block.

4.45.1.1.2   InsertLayer

The `InsertLayer` element selects a layer of a product that is to be drawn on a sheet of the drawing. The attributes of the `InsertLayer` element are defined as follows:

Attributes	Requirement	Description
productRef	REQUIRED	A reference to a Product/@id.
layerSingleRef	REQUIRED	The layer of the product that is to be drawn on the sheet.
p1clip	OPTIONAL	The lower left-hand coordinate point of the layer clipping rectangle.
p2clip	OPTIONAL	The upper right-hand coordinate point of the layer clipping rectangle.
alpha	OPTIONAL	The alpha transparency value to use when rendering the layer on the. This attribute allows different colors to be used to display multiple layers on a single drawing.
colorRef	OPTIONAL	A reference to a Color/@id attribute. The default color is black.

4.45.1.1.3   SchSymbolRef

The SchSymbolRef element creates an instance of a symbol on the drawing. The element associates the symbol with a component reference designator defined in the Components section of the GenCAM file. The attributes of a SchSymbolRef element are defined as follows:

Attributes	Requirement	Description
schSymbolRef	REQUIRED	A reference to a SchSymbol/@id.
componentRef	REQUIRED	A reference to a MountingLocation@refDes
circuitRef	REQUIRED	Associates the circuitNumber of the patter identified by Device/@patternPinRef at MountingLocation@refDes. This reference searches through the PinDesc definitions in the device referenced by componentRef and finds a matching circuitRef number. Those pins that match the circuitRef name are the pins associated with the symbol. The symPinRef value from PinDesc can then be mapped to the SymPin/@id in the symbol definition.
colorRef	OPTIONAL	A reference to a Color/@id attribute. The default color is black.

4.45.1.1.3.1 *SymPinRef*

The SymPinRef element associates a net with a symbol pin. The attributes of a SymPinRef element are defined as follows:

Attributes	Requirement	Description
symPinRef	REQUIRED	A reference to the symPinName of SchSymbol/@id.
netRef	REQUIRED	A reference to a Route/@id.

4.45.1.1.4   *Net*

The Net element associates a net with a graphic representation of the net on the drawing. The attributes of a Net element are defined as follows:

Attributes	Requirement	Description
netRef	REQUIRED	A reference to a Route/@id.

The polyline elements and text elements that follows this net element define the graphic on the drawing that are associated with the net name. These elements would be located within the Frame of the drawing and they should connect SymPin images that correspond to the same netRef.

4.45.1.1.5   *DrawGroup*

The DrawGroup element is a container for related freehand graphics and text that are to be drawn on the sheet. The DrawGroupRef inside the DrawGroup allows nesting of DrawGroups. Recursive nesting is not allowed. All elements must be defined before referenced.

Attributes	Requirement	Description
drawGroupId	REQUIRED	The identifier for the DrawGroup. The name must be unique within the drawgroup namespace within the GenCAM file. If the CAD system does not define drawgroup names then the GenCAM writer must supply names such as "dg:1", dg:2".
drawGroupRef	REQUIRED	A reference to a DrawGroup/@drawGroupId. The referenced DrawGroup must be defined prior to being refernced.

4.45.1.2   ExternalDrawing

Attributes	Requirement	Description
url	REQUIRED	The Internet World Wide Web identifier (Uniform Resource Locator) of the drawing.
drawingFormat	REQUIRED	The MIME type format for the drawing information in accordance with IETF standards RFC-2045 through RFC-2049. For example, application/IPC.GenCAM, application/postscript, or text/html. The default format is application/IPC.GenCAM.

4.46   TestPins

The TestPins element defines a list of all TestPin elements used within the GenCAM file.
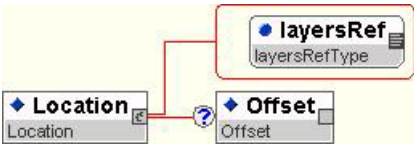
4.46.1   TestPin

The TestPin element names a pin in the test fixture. The only attribute of a TestPin element is defined as follows:

Attributes	**Requirement**	**Description**
id	REQUIRED	The id attribute is a qualifiedName that uniquely identifies the TestPin within the GenCAM file.

4.46.1.1 Location

The Location element defines a layer of a fixture starting with the layer closest to the tester and ending with the layer closest to the board, panel, or assembly under test. Multiple locations for a TestPin are defined so that if pins pass through layers at an angle they can have different locations defined on each layer. The attributes of a Location element are defined as follows:

Attributes	**Requirement**	**Description**
layersRef	REQUIRED	A reference to a LayerSingle/@id attribute or a LayerSet/@id attribute of the layers. The layer or layers through which the hole is cut.

4.47   TestProbes

The TestProbes element defines a list of all TestProbe elements used within the GenCAM file.
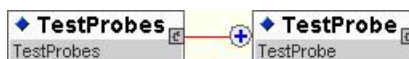
4.47.1   TestProbe
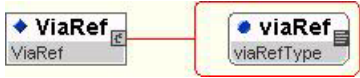
The TestProbe element defines the characteristics for the test probe. The attributes of a TestProbe element are defined as follows:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `TestProbe` within the GenCAM file.
productInstanceRef	OPTIONAL	A reference to a Placement/@productInstanceId that identifies the specific instance of a product type that is being probed. The productInstanceRefType must match "[a-zA-Z][a-zA-Z0-9_-]*".
type	REQUIRED	The type of the probe. One of: ICT - in-circuit test DUALSTAGE TRANSFER ROBOTIC - a movable, or "flying" probe
recepticalSize	REQUIRED	the probe receptacle diameter
tipSize	OPTIONAL	the testprobe tip maximum cross section dimension (diameter if round).
spring	REQUIRED	The receptacle spring force in units consistent with system of units defined in Units/length. For instance, if the length attribute is MM or UM the force is in units of Newtons. If the attribute is INCH then the force is in units of ounces.
receptDepth	REQUIRED	The recess (negative value) or raise (positive value) to be applied to the receptacle insertion.
layerSingleRef	REQUIRED	A reference to a LayerSingle/@id attribute that selects the surface of the fixture probe plate into which the probe is inserted. The `surface` attribute for the layer must be defined as TOP or BOTTOM.

tipType	OPTIONAL	The probe type for ICT or DUALSTAGE probes. One of: SPEAR CHISEL CROWN TULIP4 TULIP3 CASTLE RADIUS OTHER - an unknown or non-standard probe type.
attach	OPTIONAL	the manner in which the probe is connected to the tester. One of: CAPACITIVE INDUCTIVE MATING OTHER - an unknown or non-standard means of connection.
productInstanceRef	OPTIONAL	A reference to a Product/@productInstanceId. Select one instance of a product from a panel. The productInstanceRefType must match "[a-zA-Z][a-zA-Z0-9_-]*".

4.47.1.1 ViaRef

The `ViaRef` element refers to a `Via` on a board, panel, or assembly to be accessed for test. The only attribute of a `ViaRef` element is defined as follows:

Attributes	Requirement	Description
viaRef	REQUIRED	A reference to a VIA/@id attribute.

4.47.1.2 TestPadRef

The `TestPadRef` element refers to a `TestPad` on a board, panel, or assembly to be accessed for test. The only attribute of a `TestPadRef` element is defined as follows:

Attributes	Requirement	Description
testPadRef	REQUIRED	A reference to a TestPad/@id attribute.

4.47.1.3 ComponentPinRef

The `ComponentPinRef` element refers to a component pin on a board, panel, or assembly to be accessed for test. The attributes of a `ComponentPinRef` element are defined as follows:

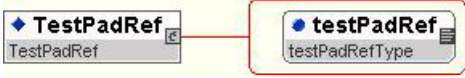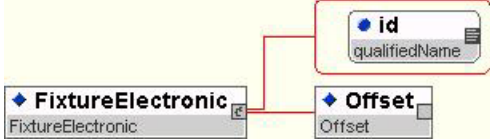| Attributes	Requirement	Description
componentPinRef	REQUIRED	A reference to a `MountingLocations/MountingLocation/@refDes` attribute and to the corresponding `ComponentPin/@mountingLocationRef` attribute. The physical location of the ComponentPin is defined in the mounting location. The Route containing the ComponentPin defintion determins the net connectivity of the pin.
patternPinRef	REQUIRED	The `patternPinRef` is only optional for non-contact probes, such as inductive probes.

4.48   FixtureElectronics
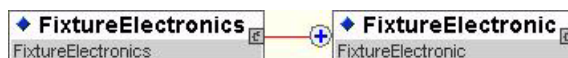
4.48.1   FixtureElectronic

The FixtureElectronics element defines a fixture-electronics terminal. For example, in the case of a testconnect tied to a tester pin this would be an input to the fixture electronics circuitry while when associated with a test probe this would be the output from the fixture electronics.

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `FixtureElectronic` within the GenCAM file.

4.49   TestConnects

The `TestConnects` section defines the tester resource connections between the board or panel, optional fixture-electronics, and the test system. The placement of the test probes can be done manually or automatically. Test connections can be on component pins, connector pins,

test pads, or vias. The following XML defines the elements allowed in the `TestConnects` section, and the constraints on their use:
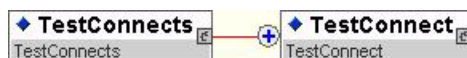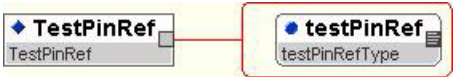
4.49.1   TestConnect

The `Testconnect` element defines an interconnection between the test system and the test fixture or the product under test The connections are generally made using wire-wraps or crimped wires between some combination of a test probe, a tester resource pin, and a fixture-electronics signal. Fixture-electronics is used here to describe any electronic circuitry which is neither part of the test system nor part of the board under test, but which is connected to one or both. The XML indicates that zero to two of `TestPinRef`, `FixElectRef`, and `FixElectRef` are allowed after a `TestConnect` element. The requirement is to have exactly two elements after `TestConnect`. The `TestConnect` is defining a connection between any combination of two of these connection points. An alternative XML to describe the relationship is:

Attributes	Requirement	Description
id	REQUIRED	The `id` attribute is a `qualifiedName` that uniquely identifies the `TestConnect` within the GenCAM file.
bunchLabel	OPTIONAL	the name used to associate several connections with each other for testing purposes (typically a cable name). All TestConnects referencing the same bunchName will be associated in the manner prescribed by the bunchType.
bunchType	OPTIONAL	a connection bunchType is either TWISTED, CABLED or BUSSED. A TWISTED type bunch must be referenced in exactly two (2) TESTCONNECT elements, each referencing a different signal name. CABLED and BUSSED type bunches must be referenced in at least two (2) TESTCONNECT elements, each referencing a different signal name.

4.49.1.1   TestPinRef

The `TestPinRef` element defines one end of a test connection within the tester. The other end could be a testprobe, a fixture electronic signal, or another testpin. The only attribute of a `TestPinRef` statement is defined as follows:
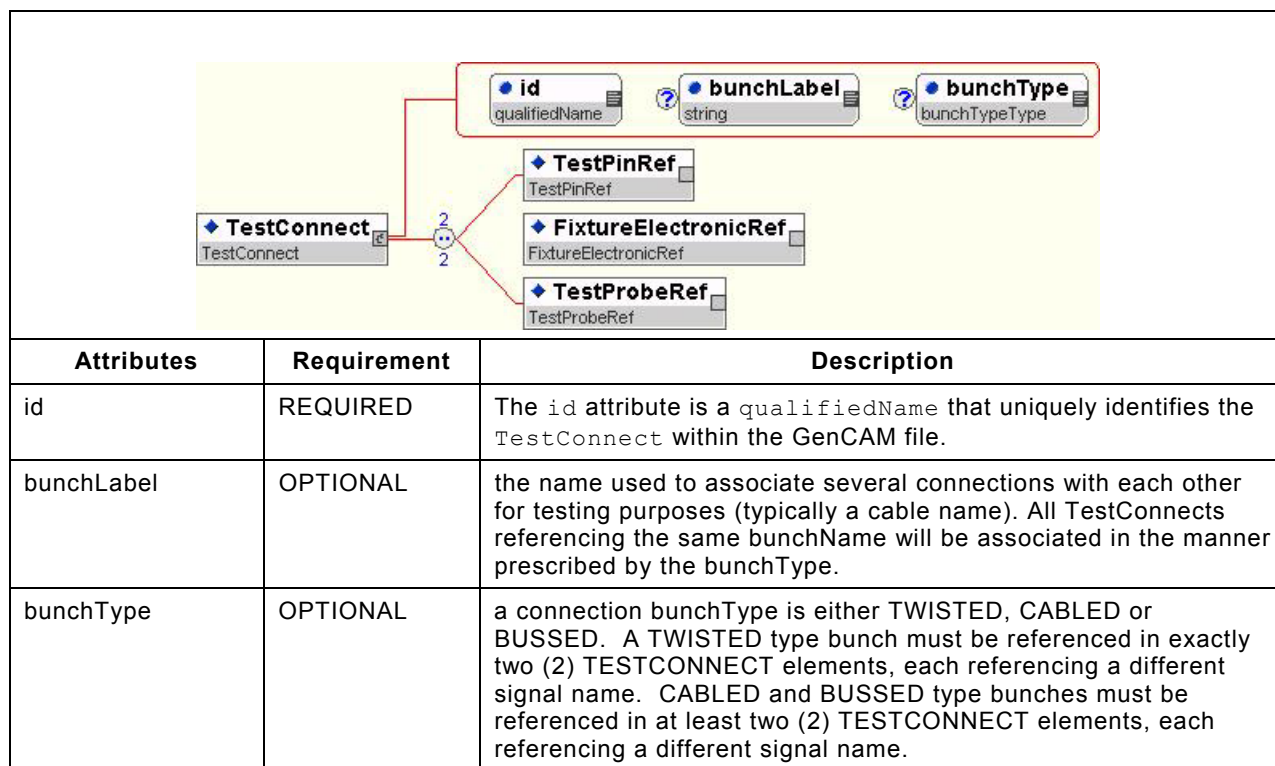
166

Attributes	Requirement	Description
testPinRef	REQUIRED	A reference to a `TestPin/@id attribute`.

4.49.1.2 *FixtureElectronicRef*

The `FixtureElectronicRef` element defines one end of a test connection within the tester. The other end could be a testprobe, a testpin. The only attribute of a `FixtureElectronicRef` element is defined as follows:

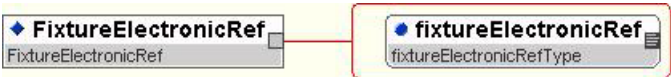| Attributes	Requirement	Description
fixtureElectronic Ref	REQUIRED	A reference to a `FixtureElectronicRef/@id attribute`.

4.49.1.3 *TestProbeRef*

The `TestProbeRef` element defines one end of a test connection within the tester. The other end could be a testpin, a fixture electronic signal, or another testprobe. The only attribute of a `TestProbeRef` element is defined as follows:

Attributes	Requirement	Description
testProbeRef	REQUIRED	A reference to a `TestProbe/@id attribute`.

4.50   Changes

The `Changes` section defines a mechanism for applying local changes to the content of the GenCAM element definition. All changes are defined relative to the GenCAM element. This provides an opportunity for those who receive the file from the original owner to indicate changes to be made to a particular element or attribute of the GenCAM data.

167

4.50.1   Revision

The `Revision` element references the `History/@number` for the file to which the changes contained in the Revision are to be applied:

Attributes	Requirement	Description
historyNumber	REQUIRED	A period separated series of numbers that indicates the revision of the file to which the changes are to be applied.

4.50.2   Change

The `Change` element define a specific action to be applied to an element or attribute in the GenCAM object. The definition of the `Change` element is define as follows:
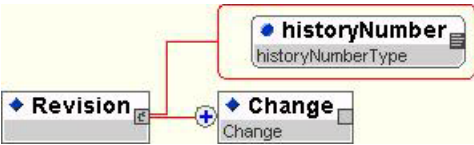

Attributes	Requirement	Description
action	REQUIRED	The type of action to be applied to the target element or attribute. If the action is DELETE or RENAME the Change element will be empty. If the action is ADD or REPLACE the contents of Change must be of a type that would be allowed at the location selected by the target XPATH. The possible action are: ADD – Add the contents at the location referenced by target DELETE – Delete the contents at the location referenced by target RENAME – RENAME the contents at the location referenced by target REPLACE – Replace the contents at the location referenced by target
target	REQUIRED	An XPath definition that selects an element or attribute within the GenCAM element to which the `Change` action applies.
timeStamp	OPTIONAL	the `xsd:dateTime` value when the edit was added to the file
personRef	OPTIONAL	The `personRef` attribute references the `Person/@id` to identify the person who made the change.

The purpose of the Changes section is to make minor changes to a GenCAM file. The number of changes is typically small relative to the file size. Large changes or additions to the file by any member of the supply chain would not utilize the `Changes`, but would make the file changes directly and increment the file `History/@increment`. A "best practice" would be to create a

168

new namespace prefix associated with the supply chain member and include this prefix in all additions.

One of the main concerns is whether the change is made or whether the change is only recorded as to what the change was. It needs to be clear between the user and the member of the supply chain as to what methodology is used for the configuration management. In some instances, the change will be automatically made to the file that is returned to the owner. In other instances, the change may be a suggestion that a manufacturer wishes to make to the file and wants permission to allow her to do that. The relationship between these conditions is established between the owner and the potential suppliers building the product.

The following is an example of a Change to a ComponentPlacementRef instance and a LineDesc instance:

```
<Changes>
   <Revision historyNumber="2.2">
          <Change action="DELETE"

target="Assemblies/Assembly/@assemblyRef['asmb1']/ComponentPlacementRefs/ComponentPlacmeentRef/@refDesRef['asm1:U6']" />
          <Change action="DELETE"

target="Assemblies/Assembly/@assemblyRef['asmb1']/ComponentPlacementRefs/ComponentPlacmeentRef/@refDesRef['asm1:R1']" />
          <Change action="DELETE" target="LineDescs/LineDesc/@id['ipcStd:signalSize0'] />
          <Change action="RENAME" from="Colors/Color/@id['red']" to="Colors/Color/@id['rust']" />
      </Revision>
      <Revision historyNumber="2.2">
          <Change action="ADD" target="LineDescs">
             <LineDesc id = "ipcStd:signalSize2" width="2.5" />
          </Change>
          <Change action = "REPLACE" target = "LineDescs/LineDesc/@id['ipcStd:signalSize1']/@width">4</Change>
          <Change    action    =    "REPLACE"   target   =   "Assemblies/Assembly/ComponentPlacmentRef/@refDesRef
['asm1:AMD1800']/>INP42100</Change>
       </Revision>
     </Changes>
```

5   REQUIREMENTS

6   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.

6.1   IPC (1)

IPC-T-50   Terms and Definitions

IPC-D-275   Design Standard for Rigid Printed Boards and Rigid Printed Board Assemblies

IPC-D-300   Printed Board Dimensions and Tolerances

IPC-D-310 Guidelines for Artwork Generation and Measurement Techniques for Printed Circuits

IPC-D-325 Documentation Requirements for Printed Boards, Assemblies and Support Drawings

6.2 American National Standards Institute (2)

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)

6.3 Department of Defense (3)

DoD-STD-100 Engineering Drawings

6.4 Electronic Industries Association (4)

EDIF 4 0 0 Electronic Data Interchange Format

6.5 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

6.6 Internet Engineering Task Force (IETF) Standards

The following IETF RFC are available from http://www.ietf.org/rfc

RFC-1738 Berners-Lee, T., Masinter, L. and M. McCahill, "Uniform Resource Locators" (URL), RFC 1738, December 1994

RFC-2045 Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, December 1996

RFC-2046 Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, December 1996

RFC-2047 Moore, K., "Multipurpose Internet Mail Extensions (MIME)Part Three: Representation of Non-ASCII Text in Internet Message Headers", RFC 2047, December 1996

RFC-2048   Freed, N., Klensin, J. and J. Postel, "Multipurpose Internet Mail Extensions (MIME) Part Four: Mime Registration Procedures", RFC 2048, December 1996

RFC-2049   Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, December 1996

Annex A

What's Changed

This section describes changes that were made to the GenCAM object model in the transition from IPC-2511A and IPC-2511B.

a) Replace the BNF based model definition and the homegrown syntax of the IPC-2511A with an XML file format defined using XML Schema. This was done to comply with the IPC-2500 series file format requirements.

b) Flattened the depth of the element tree by eliminating artificial section definitions. The HEADER section was renamed Header and many of the elements in that section were moved out to the top level of the file. Some of the old section definitions have been eliminated and their contents have been made elements directly under the GenCAM element.. The old sections that contained only one top-level keyword statement inspired this change. (Device, Mechanical, Families, Packages,Power, etc.)

c) Merged Units and AngleUnits into a single element with two attributes and set defaults to MM and DEGREES.

d) Compressed the definitions for BOARD, ASSEMBLY, PANEL, AND FIXTURE from HEADER AND ADMINISTRATION into a single element called Product.

e) Person now references an Enterprise for organizational information. Roles were introduced to allow the declarations of responsibilities to be controllably extended beyond the roles defined by IPC.

f) Eliminated the GROUP mechanism for defining namespaces. The new namespace mechanism is based on the notation used in defining namespaces for schemas. The new mechanism is more consistent with XML practices for managing names, keys and references.

g) Eliminated the STARTAT statement in the Polyline element and Polygon element. The starting point is not an attribute of Polyline and Polygon.

h) Moved xform, position, place, and offset out of element attribute lists and into child elements.

i) Made the x and y attributes in XForm, Position, Place, and Offset optional.

j) Dropped inline mechanism for instancing complex graphics TARGET, FEATURE and ARTWORK. Now all instances are created by referencing the graphics definitions.

k) Removed the DEF at the end of all definitions outside of the Primitives section.

l) Eliminated the inline definition of a PAD statement in a PADSTACK statement. The new PadStack element only allows PadRef. The PadRef references a Pad defined in the Pads section

m) Removed inline ARTWORKs. ARTWORKREFs are now the only way to place an artwork in a Pattern, SchSymbol, Mechanical, COMPONENT etc.

n) In devices the VALUE statement has been extended to match the definition defined in IPC-2547. The NumberValue is able to support ranged value and arrays of values as required during testing. An EnumerationValue was added from the IPC-2578.

o) The COMPONENT statement was split into MountingLocation and ComponentPlacement. (The COMPONENT statement had mixed board fab data with assembly data ina single statement. The split eliminates confusion on how to split up the data within namespaces.) The ComponentPlacement statement is defined relative to the MountingLocation. The MountingLocation merged the compPinRef and ConnPinRef into a single attribute, componentPinRef.

p) Added PINDESC to ComponentPlacement so that programmable devices can reflect the ability to customized the pins on a device that is attached to the board.

q) Restructured the REFERENCE statement in Route to better support standard board test procedures.

r) All layersRef attributes are now mandatory.
s) Added BoardTestDesignRules and BoardFabDesignRules to Board, and similar rules to panel
t) Moved Assembly into a separate section. Supports Assembly with Boards or Panels.

Configuration Management with GenCAM

The file `History` is defined in the `Header` element and is a manner in which users will maintain configuration and ownership of the file data. There is only one owner. That individual or facility has the responsibility for maintaining the history and is the only one permitted to increment the file master. It is understood that population of the GenCAM schema will evolve as other CAM tools will develop additional data based on the design intent.

As an example, the owner may start the history out at 1.0 where the file contains that information that comes from the CAD system necessary to define a single image. The board manufacturer who develops the fabrication details for fabricating a printed board panel, adds to the panelization element the information necessary to describe the step-and-repeat concepts for the individual board-to-panel relationship. An assembler may also add characteristics to describe the assembly panel as well as in-circuit test fixtures that are important to facilitate the building of these products.

The fabricator and assembler can increment the history by a revision letter. The owner has the responsibility for assigning that revision letter to members of the supply chain as the data moves back and forth between various suppliers. Here again, the characteristics are maintained by the owner who has the ultimate responsibility for managing the data as it arrives from different sources, or as it is passed around the supply chain. The owner should assign letters to different fabrication elements such that a fabricator might be looking at a file that they update known as "History 1.0A" while and assembler uses a file known as "1.0B History."

Although each company establishes their own criteria for configuration management, it is important to maintain the relationship between different building cycles and effectivity of when those cycles start or end. The manner in which this is accomplished is based on the determination of who is to own the file and once that is established, the configuration management characteristics can be established.

A tool that reads in a GenCAM file may filter it while the tool needs to do its work. The output, however, must be without data loss. This means that the integrity of the file must be maintained by not changing names of items that were originally read in from the source file. Whenever a file is returned to the owner, it may be with major modifications or only minor edits.

Many of the GenCAM XML elements contain an **Edit** element. The graphic representation of that element is shown in Figure 2. This graphic would be repeated throughout this document, however only the title **Edit** will be shown at later descriptions.

The purpose of the **Edit** function is to make minor changes to a GenCAM file. The number of edits are typically small, relative to the file size (usually less than 100).

Large changes or additions to the file by any member of the supply chain would not utilize the **Edit** function, but would make the file changes directly and increment the file history according to the agreements made by the user. One suggestion or practice would be to create a new namespace prefix associated with the supply chain member, and include this prefix in all

editions. An alternative might be to change the history by some sub-letter increment in order to identify those details.

IPC-2511B XML Schema

Download at: http://webstds.ipc.org