

JEDEC PUBLICATION

ECXML Guidelines for Electronic Thermal System Level Models – XML Requirements

JEP181A

(Revision of JEP181, September 2020)

NOVEMBER 2023

JEDEC SOLID STATE TECHNOLOGY ASSOCIATION



NOTICE

JEDEC standards and publications contain material that has been prepared, reviewed, and approved through the JEDEC Board of Directors level and subsequently reviewed and approved by the JEDEC legal counsel.

JEDEC 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 use by those other than JEDEC members, whether the standard is to be used either domestically or internationally.

JEDEC standards and publications are adopted without regard to whether or not their adoption may involve patents or articles, materials, or processes. By such action JEDEC does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the JEDEC standards or publications.

The information included in JEDEC standards and publications represents a sound approach to product specification and application, principally from the solid state device manufacturer viewpoint. Within the JEDEC organization there are procedures whereby a JEDEC standard or publication may be further processed and ultimately become an ANSI standard.

No claims to be in conformance with this standard may be made unless all requirements stated in the standard are met.

Inquiries, comments, and suggestions relative to the content of this JEDEC standard or publication should be addressed to JEDEC at the address below, or refer to www.jedec.org under Standards and Documents for alternative contact information.

Published by
©JEDEC Solid State Technology Association 2023
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107

JEDEC retains the copyright on this material. By downloading this file the individual agrees not to charge for or resell the resulting material.

PRICE: Contact JEDEC

Printed in the U.S.A.
All rights reserved

DO NOT VIOLATE
THE
LAW!

This document is copyrighted by JEDEC and may not be
reproduced without permission.

Organizations may obtain permission to reproduce a limited number of copies
through entering into a license agreement. For information, contact:

JEDEC Solid State Technology Association
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107
<https://www.jedec.org/contact>

This page intentionally left blank

ECXML GUIDELINES FOR ELECTRONIC THERMAL SYSTEM LEVEL MODELS – XML REQUIREMENTS

Contents

	Page
1	Scope..... 1
2	Normative References..... 1
3	Terms and Definitions..... 1
3.1	Data Terms and Definitions 1
3.2	XML Schema Key Terms and Definitions..... 2
4	ECXML Schema Definition..... 4
4.1	Units 4
4.2	ECXML Common Elements 5
4.2.1	Location..... 5
4.2.2	Size 5
4.2.3	Active 5
4.2.4	Plane 6
4.3	ECXML Top Level Elements..... 6
4.3.1	Name 6
4.3.2	Producer 6
4.3.3	Solution Domain 7
4.3.4	Materials..... 8
4.3.5	Geometry..... 10
4.4	ECXML Geometry Elements 11
4.4.1	Solid 2D Block..... 11
4.4.2	Solid 3D Block..... 12
4.4.3	Solid Cylinder 13
4.4.4	Source Block 14
4.4.5	Source 2D Block 15
4.4.6	Printed Circuit Board 16
4.4.7	Grille 17
4.4.8	Flow Resistance..... 18
4.4.9	Assembly..... 19
4.4.10	Enclosure..... 20
4.4.11	Heatsink..... 21
4.4.12	2 Resistor Model 22
4.4.13	Rectangular 2D Fan..... 23
4.4.14	Axial 3D Fan 24
4.4.15	Monitor Point 25
4.4.16	externalMcadFile..... 26
4.5	Differences Between Authoring and Importing Tool ECXML Interpretation 28
4.5.1	Overlapping Objects..... 28
4.5.2	Definition of Loss Coefficient..... 30

Contents (cont'd)

	Page
5	ECXML Annotated Examples 32
5.1	A System Level ECXML Example 32
5.2	rotationMatrix Examples..... 43
Annex A	(informative) System Level ECXML Example Listing 45
Annex B	(informative) Differences between JEP181A and its Predecessor 47
B.1	Differences between JEP181A compared to JEP181 (September 2020) 47

Figures

Figure 1 — EXCML System Level Example 32
--

ECXML GUIDELINES FOR ELECTRONIC THERMAL SYSTEM LEVEL MODELS – XML REQUIREMENTS

(From JEDEC Board Ballot JCB-23-31, formulated under the cognizance of the JC-15 Committee on Thermal Characterization Techniques for Semiconductor Packages)

1 Scope

This publication establishes the requirements for the exchange of electronic thermal system level simulation models between supplier and end user in a single neutral file format. The types of model data supported are a sub-set of what is possible to define in any one simulation tool, but with the intention of allowing model data that is common between tools to be exchanged. The data represents three-dimensional model descriptions of an electronics assembly intended to be solved so as to predict thermal and hydrodynamic behaviour using computational fluid dynamics approaches. The data is held in an XML format, conforming to an XML schema that this document describes.

2 Normative References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this publication. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this publication are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

JESD51-12, *Guidelines for Reporting and Using Electronic Package Thermal Information*.

JESD15-3, *Two-Resistor Compact Thermal Model Guideline*.

JESD15-1, *Compact Thermal Model Overview*.

3 Terms and Definitions

The following terms and definitions are applicable to this XML Schema.

3.1 Data Terms and Definitions

ECXML: ‘Electronics Cooling’ XML, named so as to indicate the scope and intention of the XML information.

Producer: The name of the tool used to author an instance of ECXML, or the name of the tool that the instance of the ECXML complies with.

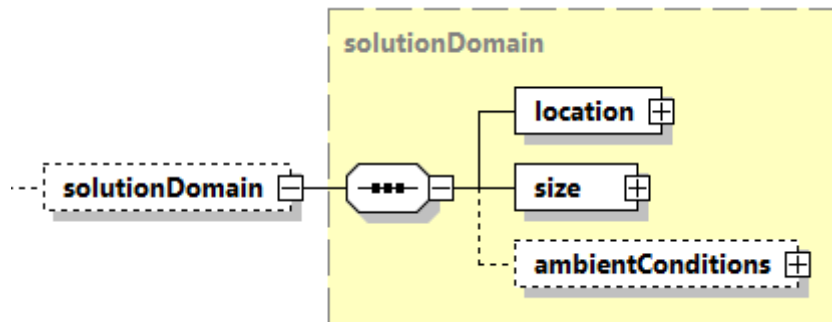
Tool: A generic name to represent any thermal simulation software that may import or export an instance of the ECXML.

Material: The physical material that an object is constructed from, including its surface properties.

3.2 XML Schema Key Terms and Definitions

An XML structure is simply a set of rules to define how to organize text into a standard structure within a file. This structure is defined by an XML schema file (.xsd). This schema file is the XML ‘dictionary’ and is distinct from an example, or instance, of an XML file that conforms to the schema. The structure of the schema is best represented by images. Software tools called schema viewers can enable you to see this structure as well and walk up and down the structured representation of the data. XML validators can be used to check if an XML file example conforms to its schema definition, e.g., it contains only the information that the schema allows for. Such validation, often performed on import of an XML file into a software, is useful for ensuring the integrity of the XML data and thus error free interpretation of it.

The following image shows a graphical representation of one section of the ECXML schema. A description of this image is provided so that all subsequent image descriptions of the schema may be understood. The XML element name will be indicated in *italics* in this document.



The line to the left of the *solutionDomain* element indicates that this element has a parent element. The dotted line box around *solutionDomain* indicates that this element is optional. If *solutionDomain* is included in the XML, then a *location* and *size* must also be included, but its *ambientConditions* definition is optional.

An element is a named structural entity in the XML. Using the *solutionDomain* example, this would be represented in the XML instance file as:

```
<solutionDomain>
  <location>
    <....>
  </location>
  <size>
    <....>
  </size>
</solutionDomain>
```

where represents the additional elements required by the *location* or *size* elements.

3.2 XML Schema Key Terms and Definitions (cont'd)

Many elements will contain a type. Simple types convey valuable information about what you are allowed to put under the element, summarised in the following table:

Type	Description
xs:string	A string containing characters, line feeds, carriage returns and tab characters
xs:boolean	Either true, false, 1 (which indicates true) or 0 (which indicates false).
xs:double	64-bit floating-point number
xs:integer	Numeric value without a fractional component

Elements with types beginning with xs: are standard available types that limits the format of the data, and hence improves the integrity of the data within the xml. Subsequent descriptions of ECXML elements will indicate what types are to be used.

The graphical representation of the schema presented in this guideline uses the following symbols to indicate the number and relationships of the XML data.

Graphic Descriptor	Type	Description
	Cardinality	Shown below an element. The cardinality defines if the element can be repeated or not. In this case there must be at least 0 and as many as needed. When 0 is the first value it means that this element is optional and can be excluded. When there is no cardinality there should only be one element of that type under the parent element.
	Sequence	Any of the elements to the right of the sequence can be added under the element to the left of the node. A sequence further retrains the XML by requiring the element be provided in the order specified by the schema which is also the order shown in the images.
	Choice	Only a single element to the right of the choice can be added under the element to the left of the node.
	All	All elements to the right of the node must be added under the element to the left of the node, unless the element to the right is optional. The node “All” reduces the constraints enforced by Sequence, insofar that there is no restriction on the order of the element within the xml file.

4 ECXML Schema Definition

The ECXML schema is described by use of graphical representations of the hierarchical order of the XML elements.

4.1 Units

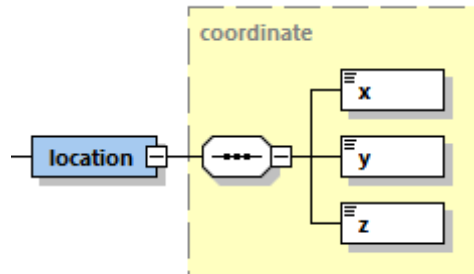
All numeric values are implicitly assumed to be in SI units. The schema does not allow for specification of a non-SI unit. The following table lists the SI units that are used by the ECXML schema:

Unit Type	SI Unit
Location/Size	Meter (m)
Temperature	Kelvin (K)
Pressure	Pascal (Pa)
Power Dissipation	Watts (W)
Thermal Conductivity	(W/mK)
Thermal Resistance	(K/W)
Specific Heat	(J/kgK)
Density	(kg/m ³)
Temperature dependent thermal conductivity coefficient	(W/mK ²)
Flow Rate	(m ³ /s)
Free area ratio, Loss Coefficient, Surface emissivity	Dimensionless

4.2 ECXML Common Elements

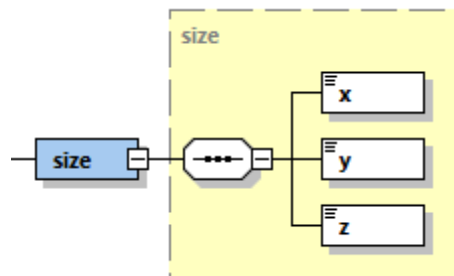
Elements that are used twice or more by other elements, e.g., low level elements, are described first.

4.2.1 Location



The *location* element defines the location (position) of an object by 3 Cartesian coordinates; X, Y and Z as `xs:double` types. These may be positive or negative values and refer to a single 0,0,0 global origin point. These 3 coordinates are the smallest X, Y, Z values of all eight corner locations of an object's bounding box. Other coordinate systems such as Polar are not supported.

4.2.2 Size



The *size* element defines the size of an object by the extents of its bounding box in the 3 coordinate directions. Size values are of positive `xs:double` types. Note that rotation of objects is not supported.

4.2.3 Active



The *active* element is an `xs:boolean` setting that indicates whether the object should be considered by a simulation or not. For the object to be active and considered, set = true or 1. For the object to be in-active and thus ignored by a simulation, set = false or 0.

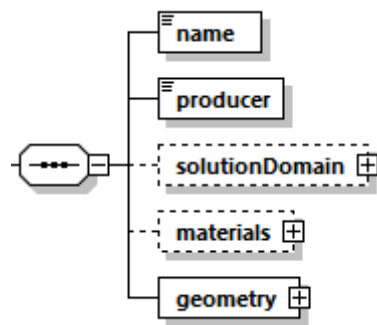
4.2.4 Plane



The *plane* element is used by two dimensional objects to indicate both an orientation and a direction (normal). The available settings are; +XY, -XY, +YZ, -YZ, +XZ, -XZ. The two letters refer to the plane orientation. The sign refers to the direction in which the object ‘points’; positive in the increasing axis direction, negative in the decreasing axis direction. Other non-Cartesian axis aligned directions are not supported.

4.3 ECXML Top Level Elements

Elements at the top level of the schema are described.



4.3.1 Name

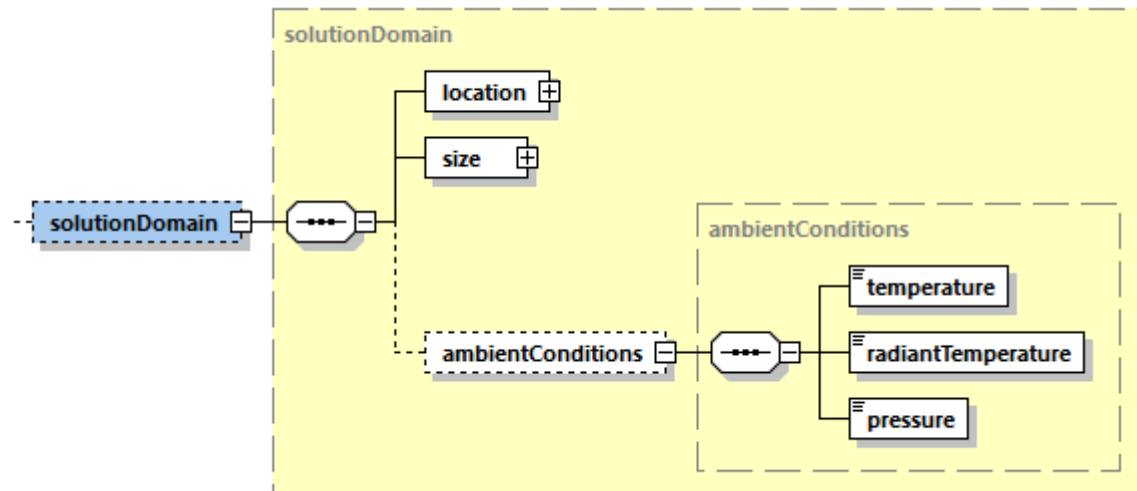
The *name* element is an xs:string that, at this level, defines the name of the model.

4.3.2 Producer

The *producer* element is an xs:string indicating the authoring tool. The values available are defined in the schema. Indication of which tool authored the XML, or what tool the XML is compliant with, is important when determining how to interpret certain situations such as when geometry is defined as overlapping (see 4.5).

4.3.3 Solution Domain

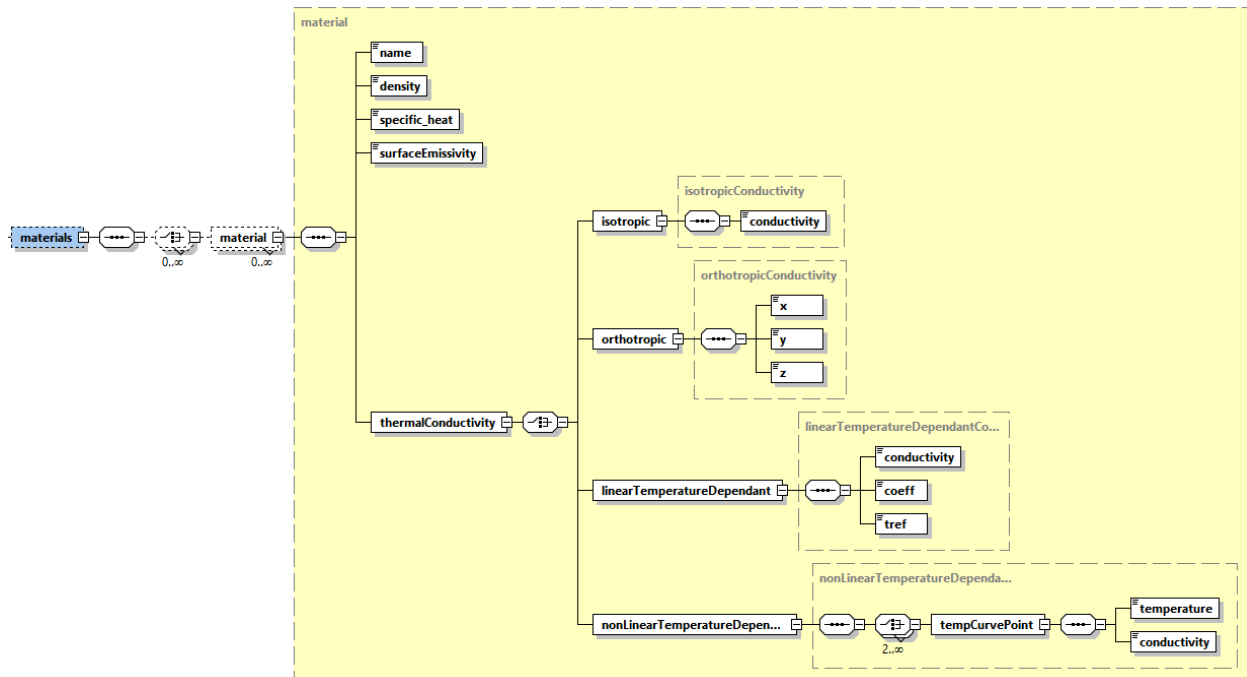
The *solutionDomain* element describes the computational domain within which the model geometry is defined.



This is an optional element. Electronics thermal models that represent just a part, or sub-assembly, need not have a *solutionDomain* defined in the XML file. If *solutionDomain* is defined then *location* and *size* are required. The *ambientConditions* values, representing the condition outside of the computational domain, are optionally specified. These are the external ambient *temperature*, an external *radiantTemperature* that is used for radiative exchange of heat from within the computational domain and the external ambient gauge *pressure*. All 3 types being xs:double.

4.3.4 Materials

The optional *materials* element holds a list of zero or more *material* elements that can be referenced by (attached to) relevant model objects.



For each *material*, a *name* is required as well as a definition of *density*, *specific_heat*, *surfaceEmissivity* and *thermalConductivity*.

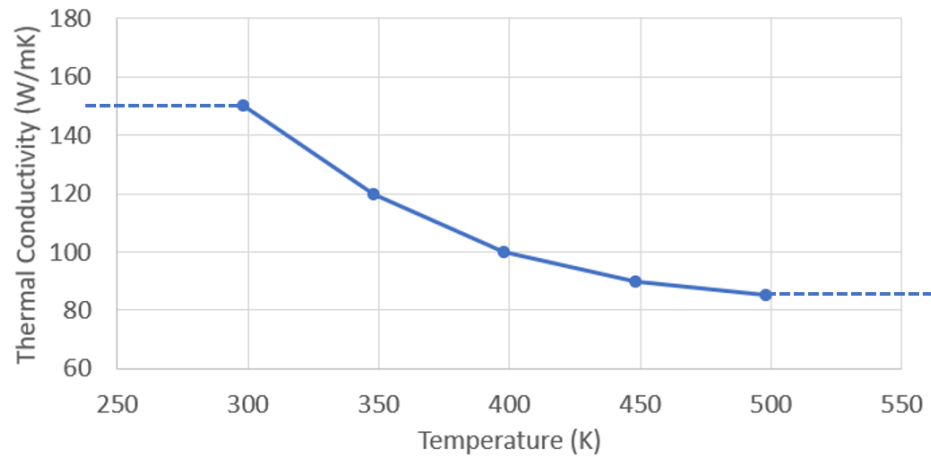
density and *specific_heat* are of type *xs:double*. *surfaceEmissivity* is of type *xs:ratio*, i.e., a value between 0.0 and 1.0.

There are 4 options for the definition of thermal conductivity:

- *isotropic*. A single value of type *xs:double*
- *orthotropic*. 3 values, each of type *xs:double*, representing the thermal conductivity in each of the 3 coordinate directions.
- *linearTemperatureDependent*. A linear relationship between thermal conductivity and temperature defined by: thermal conductivity = **conductivity** + **coeff** (Temperature – **tref**). All 3 values defining this linear relationship are of type *xs:double*
- *nonLinearTemperatureDependent*. A piece-wise linear definition of the relationship between temperature and thermal conductivity. Defined by 2 or more *temperature/conductivity tempCurvePoint* point pairs of type *xs:double*.

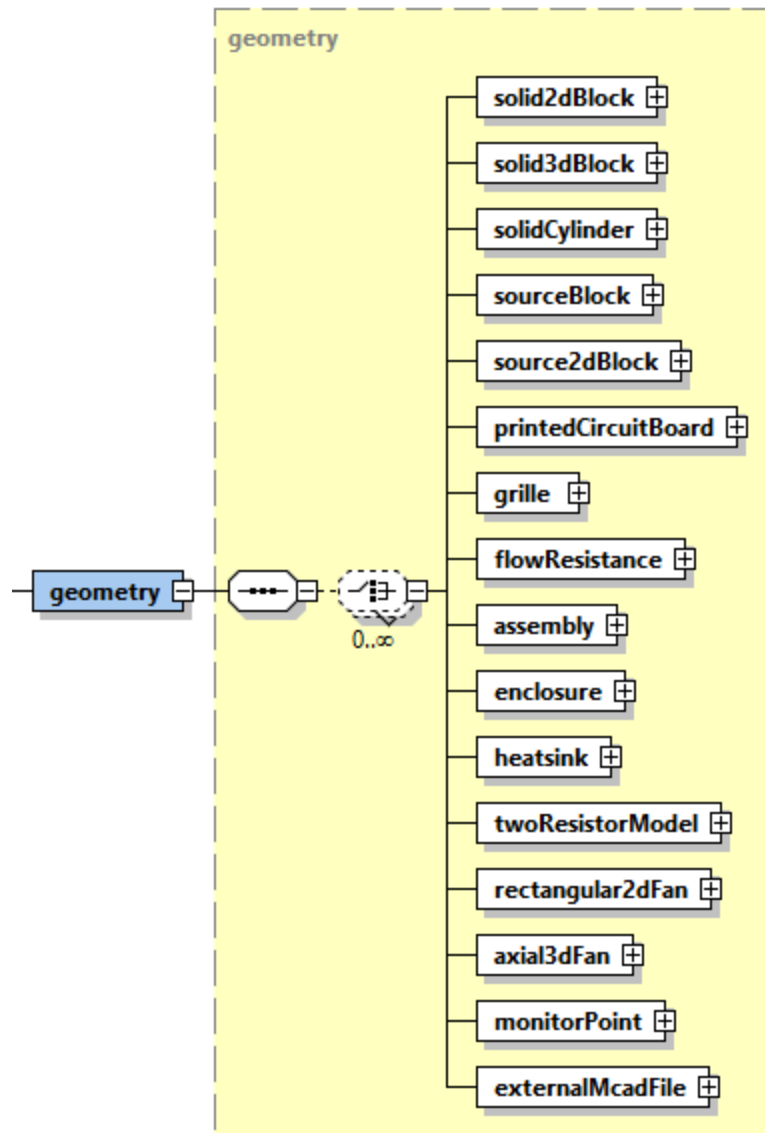
4.3.4 Materials (cont'd)

The 2 extrema points of a *nonLinearTemperatureDependent* description should define a range big enough so that expected operation occurs within it. If operation occurs outside of this range, then the temperature dependent thermal conductivity is assumed constant beyond the 2 extrema points:



4.3.5 Geometry

The *geometry* element holds a list of all model elements that go to make up construction of the system level thermal model.



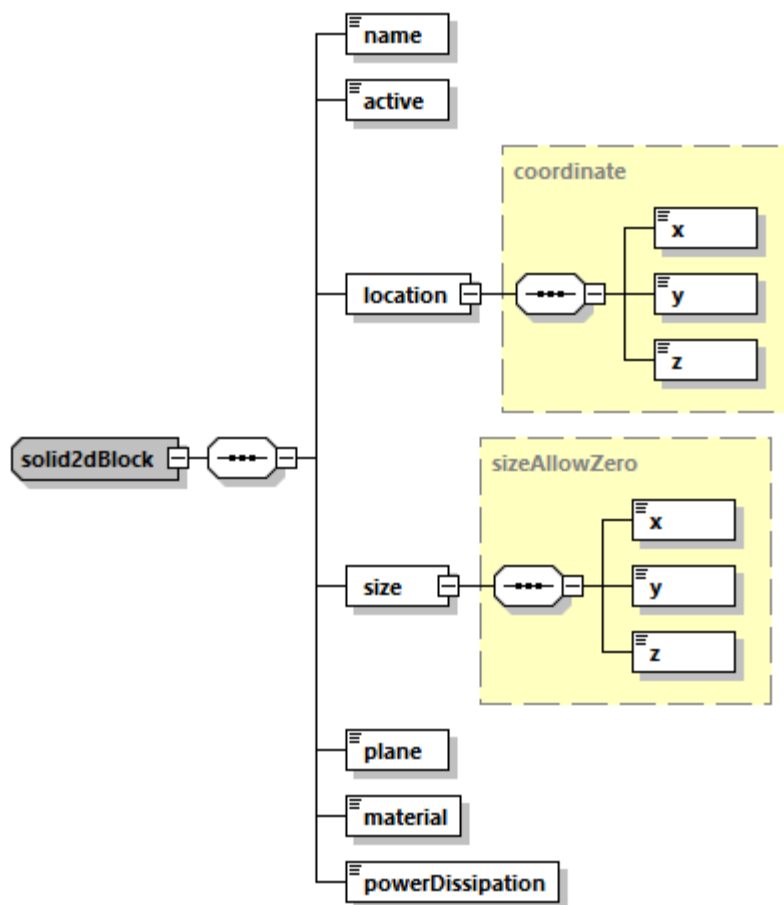
Zero or more child model objects may be present in the XML file but the parent *geometry* element must always be present.

4.4 ECXML Geometry Elements

Each model geometry element that can be defined are now described, with reference to any preceding object that has a similar data structure.

4.4.1 Solid 2D Block

The *solid2dBlock* element defines a solid object that is to be represented as a two dimensional ‘plate’ or ‘sheet’ in the simulation.



It has a *name* and an *active* setting.

It is envisioned that one of the 3 sizes is considered small enough in relation to the other 2 to the extent where the object will be considered 2 dimensional by a simulation, in the specified plane. For example if the Z size was by far the smallest compared to X and Y size, the plane setting would contain XY. That Z size may even be set to 0.0 as the X, Y and Z sizes are of an xs: sizeAllowZero type that allows for values greater than or equal to zero.