

JEDEC PUBLICATION

Part Model Guidelines for Electronic- Device Packages – XML Requirements

JEP30E

(Revision of JEP30D, August 2023)

August 2024

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 2024
3103 10th Street North
Suite 240S
Arlington, VA 22201

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.

For information, contact:

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

This page intentionally left blank.

PART MODEL GUIDELINE FOR ELECTRONIC-DEVICE PACKAGES - XML REQUIREMENTS

Contents

	Page
Foreword	iii
Introduction	iii
1 Scope	1
1.1 Purpose	1
2 Applicable Documents	2
2.1 JEDEC (www.jedec.org)	2
2.1.1 JEDEC Part Model Schema and Sub-Schemas	2
2.2 IEC (std.iec.ch)	2
2.3 IPC (www.ipc.org)	3
2.4 ISO/IEC (www.iec.ch)	3
2.5 American Mathematical Society	3
3 Terms and Definitions	3
3.1 Data Terms and Definitions	3
3.2 XML Schema Key Terms and Definitions	7
3.2.1 Sample XML Schema Structure	7
3.2.2 Sample XML Schema Types	10
3.2.3 Sample XML Schema Unique keys and Key Refs	12
4 Part Model Schema Definition	14
4.1 Part Model	14
4.1.1 Schema Release and Versioning	16
4.2 Business Info	18
4.2.1 Contact Type	21
4.2.2 Request Company	22
4.2.3 Supply Company	23
4.3 Manufacturer – Array	24
4.3.1 Manufacturer Identity	25
4.4 Company Type	25
4.5 Manufacturer Part Number – Array	26
4.5.1 Part Number Series	27
4.5.1.1 Field – Array	29
4.5.1.2 Field Code Constraints – Array	33
4.5.2 Orderable Part Number	38
4.5.3 Future Part	40
4.5.4 Standards Identifier	42
4.5.5 Linking the Manufacturer to the Manufacturer Part Number	43
4.6 Part Details - Array	44
4.6.1 Parts Selection - Array	45
4.6.1.1 Family Selection - Array	47
4.6.1.2 Parts Selection Support Contact	48
4.6.2 Association - Array	49
4.6.2.1 Part Origin - Array	50
4.6.2.2 Reference Document - Array	50
4.6.2.3 Assembly Process Classification - Array	51
4.6.2.4 Electrical - Array	52
4.6.2.5 Environmental Declaration - Array	55
4.6.2.6 Package - Array	56
4.6.2.7 Supply Chain - Array	57
4.6.2.8 Thermal Family - Array	59
4.6.2.9 Customer - Array	59
4.7 Part Origin – Array	60
4.7.1 Part Origin	60
4.7.1.1 Component Process Origin	61

PART MODEL GUIDELINE FOR ELECTRONIC-DEVICE PACKAGES - XML REQUIREMENTS

Contents (cont'd)

	Page
4.7.1.1.1 Origin Location - Array	62
4.7.2 Component Suppliers - Array	63
4.7.2.1 Company Location	63
4.7.2.2 Site Location Identity	64
4.8 Reference Manufacturer Part Number – Array	65
4.9 Reference Documents - Array	67
4.9.1 Document	68
4.9.1.1 Document Classification	70
4.9.1.2 Environmental Section	71
Annex A Workaround for tools that fail to consume Key Refs	72
Annex B (informative) Differences between JEP30 and its predecessors	73

Tables

Table 1 - XML Schema Key Definition	11
Table 2 - Schema Versioning	17
Table 3 - Part Number Ordering Information	28

Figures

Figure 1 - Part	4
Figure 2 - Alternative Manufacturer Part Numbers	4
Figure 3 - Alternative Suppliers Part Numbers	4
Figure 4 - Alternative Parts	5
Figure 5 - EDA Part	5
Figure 6 - Part Mapped to Multiple EDA Parts	6
Figure 7 - Parts Relationship to a Single EDA Part	6
Figure 8 – Examples of Combinations Allowed and Not-allowed	34

Foreword

This standard establishes requirements for the generation of electronic digital part models for electrical and electronic products for the JEDEC Solid State Technology Association. The standard defines a digital secured framework, that enables component manufacturers to submit the respective part content in a secured software consumable format that can be digitally signed off, providing Trust and Traceability to its original source. This standard specifically focuses on the parental structure, under which several sub-sections defined by their respective standards, can be contained and linked together within the Part Model parent structure. The requirements herein are intended to ensure that such part information is presented in as uniform a manner as practicable.

Where applicable, references are made to standardization documents of the following organizations throughout the Part Model hierarchy:

- American Mathematical Society.
- American National Standards Institute, Inc. (ANSI)
- Association Connecting Electronics Industries (IPC)
- Chips Alliance
- Electronics Components Industry Association (ECIA)
- Electronic Industries Association (EIA)
- High Definition Multimedia Interface (HDMI)
- Institute of Electrical and Electronics Engineers (IEEE)
- International Electrotechnical Commission (IEC)
- International Committee for Information Technology Standards (INCITS)
- MIPI Alliance (mipi)
- National Institute of Standards and Technology (NIST)
- Optical Internet Working Forum (OIF)
- Open Compute Project (OCP)
- Peripheral Component Interconnect Special Interest Group (PCI-SIG)
- SD Association (SDA)
- System Management Interface Forum (SMIF)
- The American Society of Mechanical Engineers (ASME)
- Universal Chiplet Interconnect Express (UCIe)
- Universal Serial Bus Implementers Forum (USB-IF)

These JEP30 standards represent a standard XML structure to enable Component Manufacturers to provide part data to their customers, utilizing these definitions herein in a digital model.

Introduction

This standard provides a structured part identity framework that enables the individual technical to be connected via their sub-sectional schemas as defined in the array of JEP30 related publications.

This page intentionally left blank.

PART MODEL GUIDELINE FOR ELECTRONIC-DEVICE PACKAGES - XML REQUIREMENTS

(From JEDEC Board Ballot JCB-24-29, JCB-23-33 formulated under the cognizance of the JC-11 Committee on Mechanical Standardization.)

1 Scope

This standard establishes the requirements for exchanging part data between part manufacturers and their customers for electrical and electronic products. This standard applies to all forms of electronic parts. It covers several sub-sections such as assembly process classification, electrical, environmental including material deceleration, physical, thermal, and supply chain data along with materials and substances that may be present in the supplied product or sub-products. This Guideline specifically focuses on the parental structure, under which several sub-sections listed above, can be contained and linked together within the Part Model parent structure.

All releases of any sub-schema must be under the umbrella of the Part model Schema to ensure that the Part Model schema is referencing the correct version of the sub-schema. This will enable the sub-schema to connect into the identification of the Manufacturer Part Number and the Manufacturer of the Part.

Example of how this standard can be used, is in defining the part in sufficient detail to enable process efficiencies during the part and product life cycles, i.e., design, purchasing, manufacturing, quality control, test, etc. This release includes the standardization of part data within a structured framework, to provide this support to the industry. The standard is designed to be scalable insofar that it should cover as many components as possible that are available in the market. It should also be scalable to encompass the emergence of new packages in the future.

Although this standard is considered to have international standardization implications, a complete comparison between the JEDEC standard and the international documents has not been made.

This standard applies to business-to-business transactions. The standard is not a compliance guide. As revisions to various Regulatory Regulations are released, this standard will be updated.

1.1 Purpose

This standard is intended to benefit part manufacturers and their customers by providing consistency and efficiency to the transfer of part data from part manufacturer to customers. It establishes standard electronic data exchange formats that will facilitate and improve data transfer along the entire global supply chain, at every stage in the product life cycle. A key aspect therefore is the structure of the content that is contained in this format, which the committee believes should be based on the following two principals:

- 1) Data that is required to be consumed by software tools, and

1.1 Purpose (cont'd)

- 2) Data that is not required to be consumed by software tools but is provided for informational purpose.

2 Applicable Documents

The following documents form a part of this standard to the extent specified herein. The revision of the document in effect at the time of solicitation shall take precedence.

2.1 JEDEC (www.jedec.org)

JEP30-A100, Part Model Assembly Process Classification Guidelines for Electronic-Device Packages – XML Requirements

JEP30-E100, Part Model Electrical Guidelines for Electronic-Device Packages – XML Requirements

JEP30-P100, Part Model Package Guidelines for Electronic-Device Packages – XML Requirements

JEP30-S100, Part Model Supply Chain Guidelines for Electronic-Device Packages – XML Requirements

JEP30-T100, Part Model Thermal Guidelines for Electronic-Device Packages – XML Requirements

JESD99, Terms, Definitions, and Letter Symbols for Microelectronic Devices

JEP95, JEDEC Registered and Standard Outlines for Solid State Products

2.1.1 JEDEC Part Model Schema and Sub-Schemas

JEP30-10, Part Model Schema

JEP30-A101, Part Model Assembly Process Classification Schema

JEP30-E101, Part Model Electrical Schema

JEP30-P101, Part Model Package Schema

JEP30-S101, Part Model Supply Chain Schema

JEP30-T101, Part Model Thermal Schema

JEP30-D10, *Part Model Schema Types Dictionary* (Required to support the Part Model Schema and each of its sectional sub-schemas.)

2.2 IEC (std.iec.ch)

IEC 62474, Material Declaration for Products of and for the Electrotechnical Industry

2.3 IPC (www.ipc.org)

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

2.4 ISO/IEC (www.iec.ch)

IPC/IEC 6523, Information technology – Structure for the identification of organizations and organization parts

2.5 American Mathematical Society

“Short Math Guide for L^AT_EX”, Version 1.09 (2002-03-22), currently available at

<http://www.ams.org/tex/short-math-guide.html>

3 Terms and Definitions

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

All definitions and terms associated with any sub-section schema are in accordance with the Guidelines for that sub-schema.

All other definitions and terms necessary to define the Part Model parent schema, are defined in this document.

3.1 Data Terms and Definitions

device: A piece of equipment, a mechanism, or another entity designed to serve a special purpose or perform a special function.

NOTE 1 In JEDEC documents, the word “device” is often used as an abbreviated reference to the type or types of solid-state devices that are within the scope of those documents. Context could indicate otherwise; e.g., in the phrase “the device used to hold the device under test”, the first usage of the word “device” refers to a mechanism; the second to a solid-state device.

NOTE 2 Contrast with “component”.

Component: A constituent part.

NOTE 1 Examples include source and drain regions as components of transistors, terminal frames and dice/dies as components of packaged integrated circuits, resistors and integrated circuits as components of printed circuit boards, motherboards as components of computers, LCD screens as components of monitors, ac and dc components of complex waveforms, and loops and algorithms as components of software programs.

NOTE 2 Unless the context identifies the thing of which a component is a part, a descriptive prepositional phrase identifying the thing should follow the word “component”.

3.1 Data Terms and Definitions (cont'd)

Part: As used within the context of the Part Model, a Part is the same as a device. Each Part has an identity. But there are many variations of these identities; hence the following definitions will clarify these variations.

NOTE A **Part** is a unique device that is manufactured by a specific manufacturer and has been assigned a specific manufacturer part number (MPN). It can only be produced by one manufacturer.

Parts
— Part is same as One Device

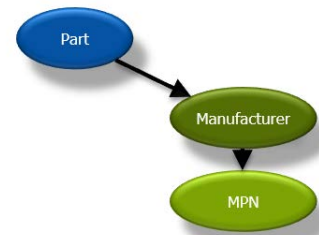


Figure 1 - Part

Manufacturer Alternative Part Numbers: When the part manufacturer ships these Parts in different supply forms, such as a reel, tube, box, or tray, the part manufacturer will assign different manufacturer part numbers to this Part to represent the part in a specific supply form. These numbers are defined as “**Manufacturer Alternative Part Numbers**”. All Alternative Manufacturer Part Numbers that represent the same Part, in different Supply Forms, must have EXACTLY the same specification, because they all represent exactly the same device.

Alternative Manufacturer Parts Numbers
— Same Part

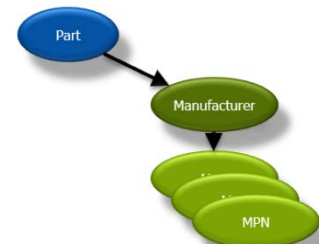


Figure 2 - Alternative Manufacturer Part Numbers

Alternative Supplier Part Numbers: When a Supplier distributes Parts or resells Parts, or acts as an Agent for a Manufacturer, the supplier usually assigns a Supplier Part Number to the Part. Just like the Manufacturer, the Supplier can have many different Supplier Part Numbers for the same Part. In addition, there can be many different Suppliers that can distribute this Part from the part manufacturer. These numbers are defined as “**Alternative Supplier Part Numbers**”.

Alternative Supplier Parts Numbers
— Same Part

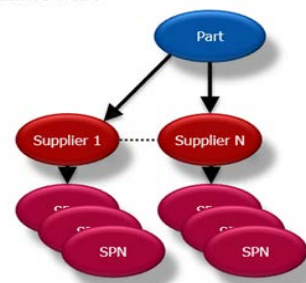


Figure 3 - Alternative Suppliers Part Numbers

Alternative Parts: To reduce risk of Product supply, multiple sources of equivalent Parts are desired, which is typically done via an Approved Manufacturers List (AML) that accompanies a Bill of Materials (BOM) for a given Product design. A Customer Part Number (CPN) or sometimes defined as an Internal Part Number (IPN) is assigned to a group of Parts that can be placed on the same Reference Designator in a Product Design.

3.1 Data Terms and Definitions (cont'd)

The Specification of the CPN/IPN represents the envelope of the specification of each **Part** that is contained within that group. All Parts connected to the same Customer / Internal Part Number are interchangeable in order for the end Product to still function as normal. These Parts are now referred to as “**Alternative Parts**”.

Alternative Parts

— Different Parts from different Manufacturer's

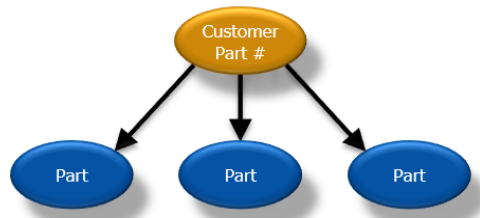


Figure 4 - Alternative Parts

EDA Part: An “**EDA Part**” represents the Design representation of a Part. It includes the following: 1) Terminal List, 2) Footprint, and 3) Schematic Symbol.

NOTE It does not include the values that can be propagated to the Schematic Symbol, as these can be inherited from the CPN/IPN as opposed to be inherited directly from the Part.

EDA Part

— A Design representation of a Part

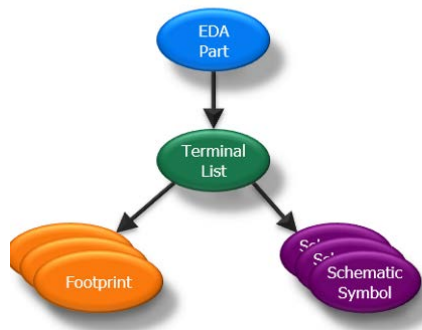


Figure 5 - EDA Part

An **EDA Part** must be connected to a **Part**. There can be many EDA Parts connected to the same Part, however all EDA Parts that are connected to the same Part must be interchangeable.

3.1 Data Terms and Definitions (cont'd)

NOTE 1 The same EDA Part can be connected to many different Parts.

NOTE 2 The same EDA Part can be reused by many different Parts, so long as the specification of these Parts maintain some commonality from the perspective of Symbol and the Footprint. This is possible since the Part specific values are excluded from the EDA Part. For example, all Resistors of different Values, Wattage and Tolerance can inherit the same Symbol. All Parts contained within the same Package can be mapped to the same Footprint.

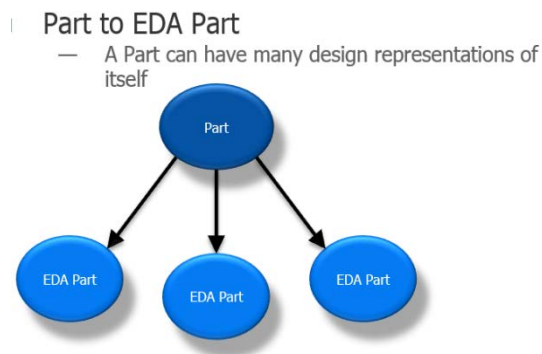


Figure 6 - Part Mapped to Multiple EDA Parts

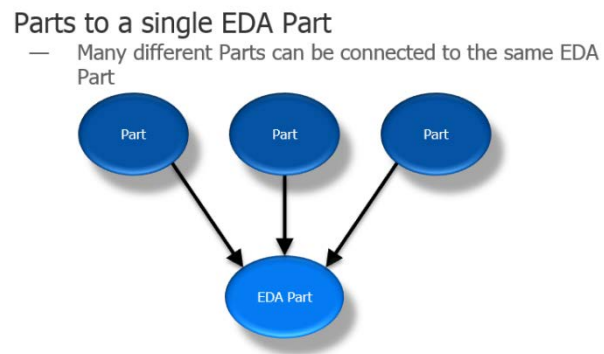


Figure 7 - Parts Relationship to a Single EDA Part

Part Model: A Part Model is a data representation described in an XML file that conforms to the rules and structure of the Part Model XML Schema.

NOTE 1 Companies who use the Part Model XML Files and claim compliance to JEDEC, must ensure that their Part Model XML file conforms to the specific released version of the Part Model XML Schema released by JEDEC.

NOTE 2 Section 4 will define the outline of the structure of the Part Model XML Schema. Specific components of the XML Schema and their hierarchy are specifically controlled by the Standards Committee who retain the expertise for these structures.

Array: The use of Arrays within the XML Schema enables a list of items to be captured under the array.

Family: The use of Family within the XML Schema means that the object can represent a family of parts each with identical characteristics. The family sections are typically referenced from another branch (typically MPN). This provided the ability of linking a large set of data (e.g. 100 data elements) that is represented once, to many other records (e.g. a 1000 MPN records) by means of an ID, instead of having to duplicate the 100 data elements a 1000 times.

3.2 XML Schema Key Terms and Definitions

XML structure is simply a set of rules to define how to organize text into a standard structure within a file. This structure is best represented by images. The key definition specified in Table 1 below helps you to understand how to interpret the images used to document the schema throughout this document. Software tools that provide schema viewing capability can enable you to see this structure as well and walk up and down the structured representation of the data. There is significant detailed information on XML that can be found online.

3.2.1 Sample XML Schema Structure

path	PartModel/ManufacturerPartNumber-Array
diagram	
type	MPN-ArrayType, ManufacturerPartNumbersType, PartNumberType, PartNumberField-ArrayType, FieldCodeConstraints-ArrayType, OrderablePartNumber-ArrayType, JEP30-D10:SignatureDigestLinkType, ds:DigestMethodType, ds:DigestValueType, SignatureType.

The *ManufacturerPartNumber-Array* is the top-level element shown in this diagram. However, you can see a horizontal line entering the element from the left-hand side, indicating that this Element has a parent. This is also denoted by the *path* and is shown here in the top line of the table as *PartModel/ManufacturerPartNumber-Array*.

The *ManufacturerPartNumber-Array* element has a *type* called *MPN-ArrayType*. This element has a structure below it, as shown by the boxes to the right-hand side of the element.

An Element is a named structural entity in the XML. This would translate inside an xml file to

3.2.1 Sample XML Schema Structure (cont'd)

```

<ManufacturerPartNumber-Array>
  <ID>MPNs ID 1</ID>
  <PartNumberSeries>
    <.....>
    < PartNumberSeries >
    < ManufacturerID>Manufacturer ID 1</ManufacturerID>
  </ManufacturerPartNumber-Array>

```

where ... can represent any number of additional elements contained within the [PartNumberSeries](#) element. The elements to the right of [ManufacturerPartNumber-Array](#) are its children such as [PartNumberSeries](#). The element to the left of [Field-Array](#) is its parent which is the [PartNumberSeries](#) element.

[ID](#) fields throughout the schema and sub-schemas are used to allow one section of the XML to reference another section of the XML. They do not have any relevance or meaning outside of the XML.

The [Field-Array](#) has a solid line meaning that it is a required element, however, it is only required if its parent is also present in the XML. This means that if any data under the [PartNumberSeries](#) element is populated, then the following data elements must be present: - [ID](#), [Name](#), [Version](#), and [Field-Array](#). The [FieldCodeConstraints-Array](#) is not mandatory, since that is an optional element as denoted by the dotted line surrounding the element. The XML is still considered valid in that it complies with the schema when optional elements are excluded, however, if the information is applicable it should be populated to benefit the User of the data. In this example, if there were no constraints that needed to be applied to the [PartNumberSeries](#) element, then there would be no need to specify any data under this [FieldCodeConstraints-Array](#) branch.

The [path](#) shown in the top row of the tables, is the path within the xml schema from the parent to the respective element in the schema. Each [path](#) is unique within the schema. Paths have limitations, insofar that there can be no spaces in the path name, and no section of the path can begin with a numerical. Non-alphanumeric characters are not-allowed except for “-” and underscores.

The [diagram](#) shows a graphical representation of the structure of the data elements within the schema. XML files that conform to this schema will have its data structured in accordance with this diagram.

Throughout the schema, there are various digital signatures such as in this example, [ManufacturerPartNumbersIdentitySignature](#), which has a [type ds:SignatureType](#). This is to provide the capability to apply a digital signature to all the content that falls in under a single representation of [ManufacturerPartNumbers](#) element. All parts listed via the unbounded element of [PartNumberSeries](#) or the unbounded element of [OrderablePartNumber](#) should belong to the same group of Manufacturers Parts Numbers (typically what is represented on a datasheet) for which this PartModel and its technical content is intended to represent.

Since the Manufacturers Parts Numbers come from a specific Manufacturer, whose details are represented in another array, a reference to that manufacturer is made via the [ManufacturerID](#), which is also accompanied by an optional [ManufacturerSignature](#) link. This signature has a type called [SignatureDigestLinkType](#) and its purpose is to bring into this array the [DigestMethod](#) and the [DigestValue](#) from the [Manufacturer-Array](#) that accompanies that [ManufacturerID](#).

3.2.1 Sample XML Schema Structure (cont'd)

While digital signatures are optional, the provision of digital signatures through the document helps the consumer validate upon receipt that the PartModel contents are not maliciously altered or tampered with. Since the PartModel structure spans so many distinct technical and marketing structures, the application of digital signatures throughout the schema offers consumers of the PartModel confidence that each top-level sections are valid at the time of consolidating into the overall and final PartModel file. These signatures also provide efficient methods for software tools to effectively join data from multiple different Part Model files into a single Part Model file for the same set of *ManufacturerPartNumbers*.

sample path	PartModel/SupplyChainSection/SupplyChain-Array/AlternativePart-Array/AlternativePart/AlternativePartSignature
diagram	
type	JEP30-D10:OptionalResponseType, ContactType, CompanyType, ds:SignatureType.

In some cases where a supplier accumulates many sections of the PartModel from different companies, the digital signature may not be adequate. In this scenario, a new *type JEP30-D10:OptionalResponseType* enables the digital signatures to be accompanied with some additional details such as authorizer, supply company and contacts for the submission of that document (or portion of the PartModel) before it is merged to form a more complete Part Model file.

An example of this could be where the Die is manufactured by a foundry who produces a file containing electrical representation data for the part. An OSAT may have outsourced the design of a package that will host that die. The OSAT or their outsourced design house may provide the package details of the PartModel. The component manufacturers may now want to merge these two sources of partial part models and join them together to provide a complete part model to their customer.

3.2.2 Sample XML Schema Types

path	PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/SurfaceTerminal
diagram	The diagram illustrates the structure of the <code>SurfaceTerminal</code> element. It shows a base class <code>SurfaceTerminal</code> with type <code>SurfaceTerminalType</code>. This class has a dashed box labeled <code>SurfaceTerminalType</code> containing several subclasses: <code>D-Shape</code> (type <code>EmptyType</code>), <code>Pullback</code> (type <code>EmptyType</code>), <code>Castellated</code> (type <code>CastellatedType</code>), <code>Hole</code> (type <code>EmptyType</code>), <code>With-opening</code> (type <code>EmptyType</code>), and <code>Open-Ring</code> (type <code>RingType</code>). The <code>SurfaceTerminal</code> class is connected to the <code>SurfaceTerminalType</code> box, which in turn connects to the subclasses.
type	SurfaceTerminalType, CastellatedType, RingType.

Many elements will contain a type. Simple types convey valuable information about what you are allowed to put under the element, such as `xs:string`, `xs:Boolean`, `xs:integer`, `xs:decimal`, `xs:dateTime`, `xs:date` etc. 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. Elements that have types excluding `xs:` represent a structure of data for that element such as an enumerated list of values, or another hierarchal structure below that element.

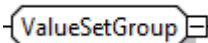
An element defined as a string expresses that a free form string must be defined with that element, such as the name of “SMT Ceramic Chip Capacitor” under the `PartNumberSeries` element would be defined as

```
<PartNumberSeries >  
  <ID>Part Construct 1</ID>  
  <Name>SMT Ceramic Chip Capacitor</Name>  
  ....
```

However, the type called `EmptyType` such as seen on the elements `D-Shape`, `Pullback`, `Hole`, and `WithOpening` simply defines the presence of that feature on the `SurfaceTerminal`. Having the type called `EmptyType` as opposed to having no type defined, allows for better recognition of the data by software tools.

3.2.2 Sample XML Schema Types (cont'd)

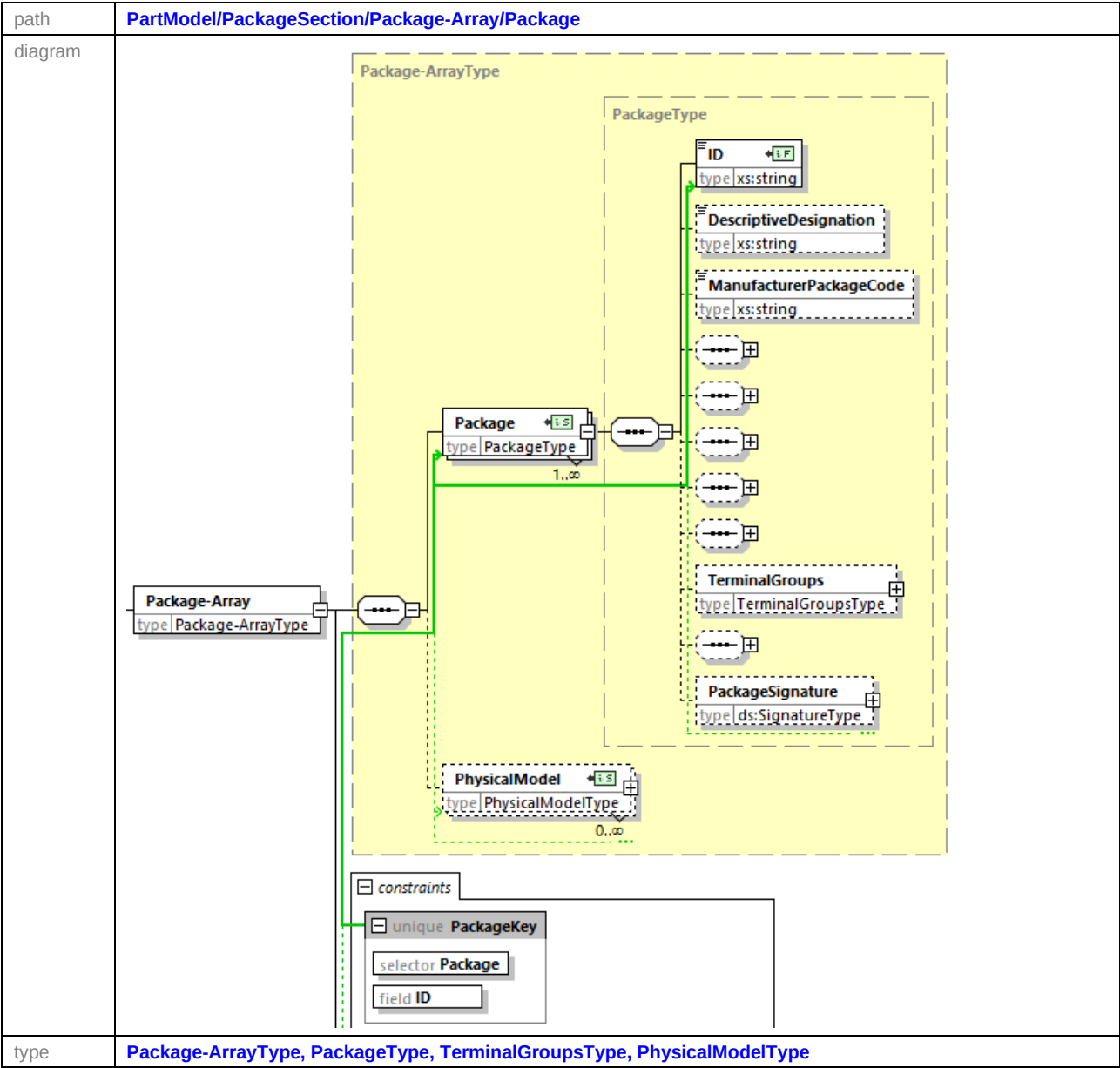
Table 1 - XML Schema Key Definition

<u>Graphic Descriptor</u>	<u>Type</u>	<u>Description</u>
	CARDINALITY	SHOWN BELOW AN ELEMENT. THE CARDINALITY DEFINES IF THE ELEMENT CAN BE REPEATED OR NOT. IN THIS CASE THERE MUST BE AT LEAST 1 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.
	GROUP	THE GROUP ELEMENT IS USED TO DEFINE A GROUP OF ELEMENTS TO BE USED IN COMPLEX TYPE DEFINITIONS

3.2.3 Sample XML Schema Unique keys and Key Refs

In addition to defining the required structure of an XML document, schemas provide mechanisms for associating identifiers to sections of the document, and for cross-referencing those identifiers from other parts of it. Identifiers are enforced to be unique, and references are validated to point to identifiers that exist.

A *unique* key constraint is the association of an identifier element to a section of the document within a scope. The following diagram shows how the schema declares that, within a *Package-Array*, a single *Package* is identified by the value of its *ID* element:



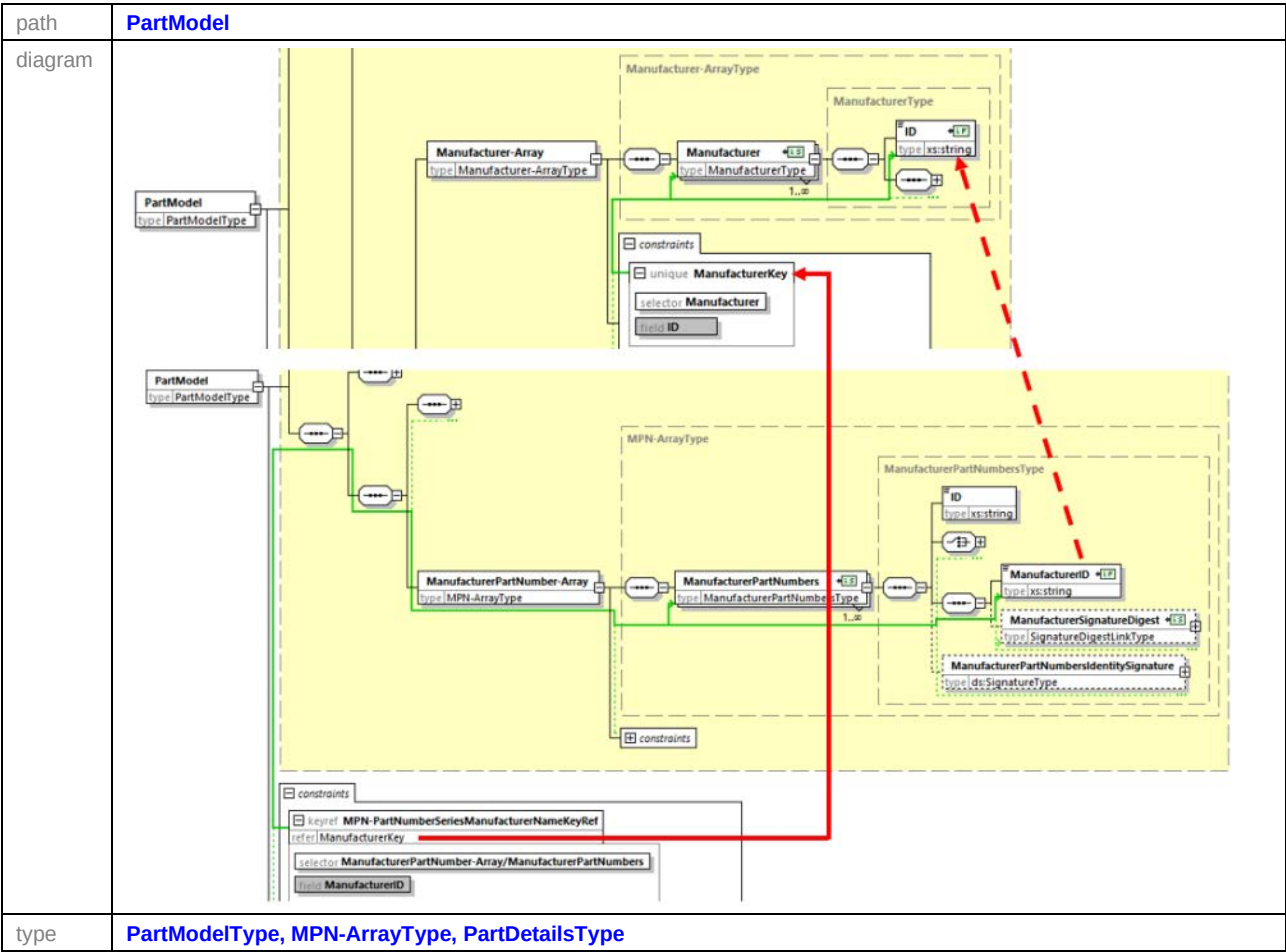
The *PackageKey* constraint enforces that no different packages from the same *Package-Array* can have the same *ID*.

3.2.3 Sample XML Schema Unique keys and Key Refs (cont'd)

A *keyref constraint* is a declaration that, within a scope, a value refers to another element in the document, which has an associated key to it.

Keyrefs must be declared at a level under which both the key and the reference to it are visible. Therefore, many keyrefs end up being declared under *PartModel*, the top-level element.

The following diagram, in which most elements have been omitted for clarity, shows how a specific *Manufacturer* within the *Manufacturer-Array* is linked to the set of parts under the *ManufacturerPartNumbers* element.

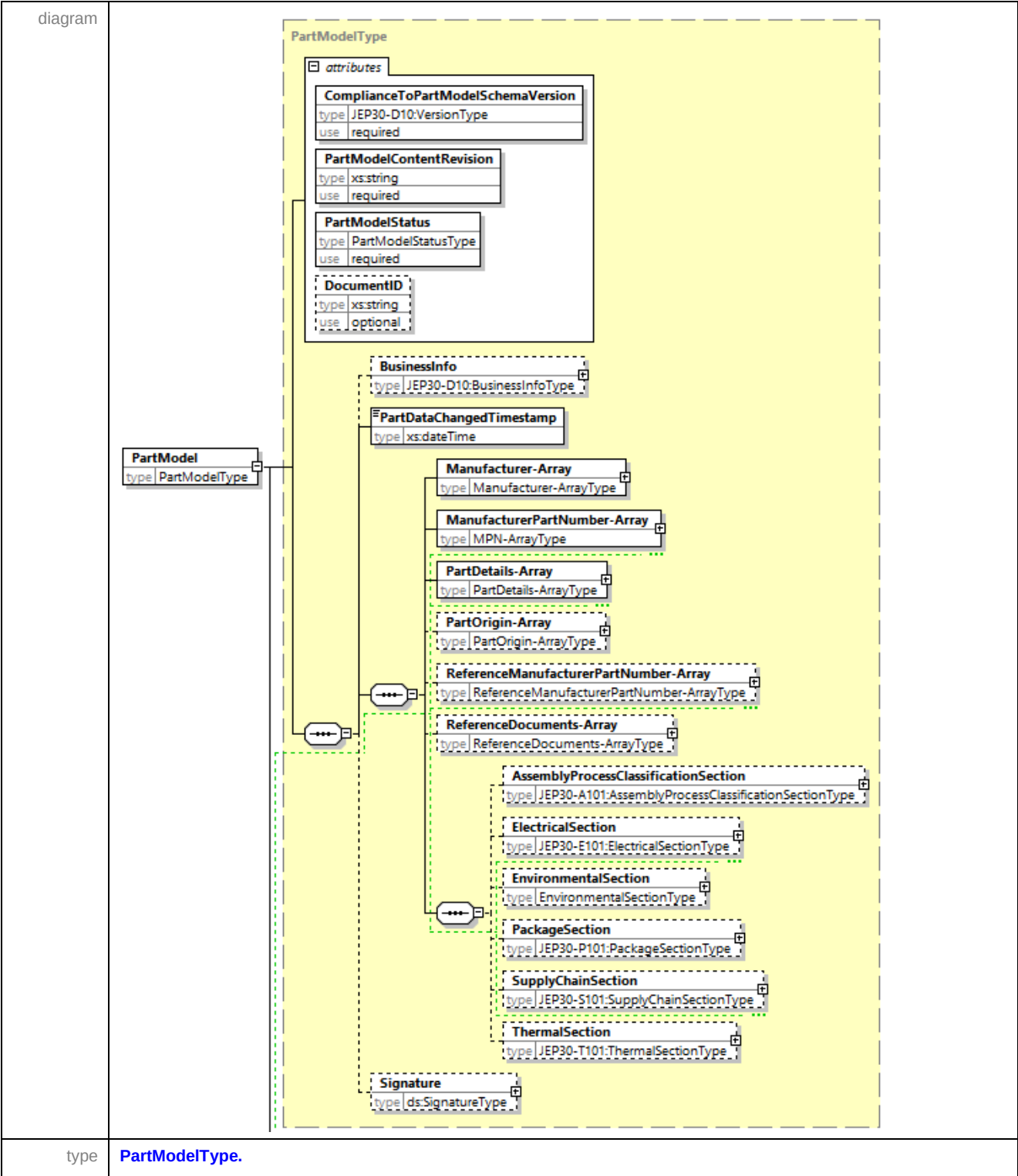


The *Manufacturer/ID* has a constraint *unique* key with the name *ManufacturerKey*. The *ManufacturerPartNumbers/ManufacturerID* has a constraint *keyref* with its own unique name, but it also includes a reference back to the key named *ManufacturerKey*. Both the key and keyref are unique within the entire schema.

4 Part Model Schema Definition

The following section describes the XML Schema structure.

4.1 Part Model



4.1 Part Model (cont'd)

The primary purpose of the Part Model Schema is to provide the structure for identifying unique parts (Manufacturer and MPNs), and the structure to include the sub schemas which define the part details. The Part Model is the high-level parent containing:-

- 1) Compliance to Part Model Schema Version
- 2) Part Model Content Revision
- 3) PartModel Status
- 4) Business Info
- 5) Part Data Change Timestamp
- 6) Manufacturer - Array
- 7) Manufacturer Part Number – Array
- 8) Part Details - Array
- 9) Part Origin - Array
- 10) Reference Manufacturer Part Number – Array
- 11) Reference Document - Array
- 12) References to Sub-schemas
 - i. Assembly Process Classification
 - ii. Electrical
 - iii. Environmental
 - iv. Package
 - v. Supply Chain
 - vi. Thermal

This document covers the parental structure of the part model, and its reference to its sub-schemas. The content under any sub-schema is tied to the Manufacturer's name and Manufacturer's part number as seen in the following section. The same concept applies to all the sub-schemas, thus enabling the data to be connected from one sub-schema to another sub-schema, to make a complete set of data for any given part or set of parts.

The [ComplianceToPartModelSchemaVersion](#) indicates the version of the Schema to which the XML file is to be validated against.

The [PartModelContentRevision](#) indicates the revision of the data for the Part that is submitted in the XML file. This enables the Component Manufacturer to provide a new XML file for a Part each time they wish to upgrade a new set of data for a part, in any of the child sections.

The [PartModelStatus](#) is a mandatory attribute that determines the status of the PartModel XML file. It has enumerated values of [Pre-Release](#), [Released](#), [Superseded](#), and [Withdrawn](#).

Both the part model Schema and each of the sub-schemas include a reference to the Part Model Types Library, which enables the xml data types to remain consistent from one sub-schema to another sub-schema.

4.1.1 Schema Release and Versioning

Release and versioning are synchronized between each sub-schema and its part model parent schema. The versioning concept follows the logic outlined below:-

- 1) The format of the version number is X.Y.Z, where
 - i. X increases when there is a change in the released standards document, that requires a change to the XML Schema to support that document, e.g.
 - a. Updates to the J-STD-075 or to the J-STD-020, that drive a change in the *“AssemblyProcessClassificationSection”* of the Part Model Schema, or
 - b. Updates to the JESD30, that drive a change in the *“PackageSection”* of the Part Model Schema, or
 - c. Updates to the JESD77, JESD99, etc. , that drive a change in the *“ElectricalSection”* of the Part Model Schema, or
 - d. Updates to the J-STD-046 or to the J-STD-048, that drive a change in the *“SupplyChainSection”* of the Part Model Schema, or
 - e. Updates to the JESD51-XX, that drive a change in the *“ThermalSection”* of the Part Model Schema, or
 - f. Etc.,
 - ii. Y increases when there is a backwards incompatible change, such as
 - a. A previous optional element now being made mandatory,
 - b. A new mandatory element being added, or
 - c. Changing the structure of a branch which breaks the backward incompatibility of the schema.
 - iii. Z increases when there is a backwards compatible change, such as but not limited to:-
 - a. A previous mandatory element now being made optional,
 - b. A new optional element being added, or
 - c. An existing element being made unbounded.

Backward incompatible changes may require clients to update their systems that use the part model.

Each time that a sub-schema gets updated, then the part model version also gets updated to release that sub-schema under the umbrella of the Part Model. This is because the Part Model must now reference the new version of sub-schema, since all sub-schemas have their own version number. The parent schema includes them by referring to a precise version, so a version bump in the sub-schema requires a version bump in the parent schema only at the time of release of the Parent.

XML Files that are used for communicating part data to the User, can be validated against any version of the *PartModel* schema to ensure that it is valid according to that Schema. Validation of a specific parts XML against the schema does not ensure correctness of content. It only ensures the structure is correct and ensures some portions of the content use an acceptable value.

4.1.1 Schema Release and Versioning (cont'd)

Table 2 - Schema Versioning

Part Model	Sub-Schemas					
	Assembly Process Classification	Electrical	Package	Supply Chain	Thermal	Part Model Types Library
1-0-0	1-0-0	1-0-0	1-0-0		1-0-0	1-0-0
2-0-0	2-0-0	2-0-0	2-0-0	1-0-0	2-0-0	2-0-0
3-0-0	2-0-1	2-0-1	3-0-0	1-0-1	2-0-1	3-0-0
4-0-0	2-0-1	3-0-0	4-0-0	1-0-1	2-0-1	3-0-0
5-0-0	2-0-2	3-0-1	5-0-0	1-0-2	2-0-2	3-0-1

The purpose of this organization is to enable the Part manufacturer to focus their data transfer to any specific sub-set of data that is required by the consumer of the data, without having to define the entire set of content applicable to the Part. This means that a part manufacturer can create two separate files containing

1. [Manufacturer-Array](#), [ManufacturerPartNumber-Array](#), [PartDetails-Array](#) details plus the [PackageSection](#) details,
2. [Manufacturer-Array](#), [ManufacturerPartNumber-Array](#), [PartDetails-Array](#) details plus the [AssemblyProcessClassificationSection](#) details.

The consumer of those files can then extract out the data from each of these files and merge them together to form a more complete data set for a specific part. Validation that the two files contain the same [Manufacturer-Array](#), [ManufacturerPartNumber-Array](#), can be done by comparing the [Digest Method](#) and [Digest Value](#) contained within the [ManufacturerSignature](#) and the [ManufacturerPartNumbersIdentitySignature](#).

Alternatively, the part manufacturer can decide to include all these branches of data into the one file. The file can support data across all the components of the xml schema, for just one Part, or for multiple Parts.

The concepts of separating out the MPN from the Family details is based on the following concepts

1. The list of MPN data can be very big.
2. There can be many Alternative MPN's that can be inserted into the file under the Array

Since many of these MPN's, and Alternative MPN's (each represented by a different string of alphanumeric characters), can map to a single XML component representation, e.g. the Package component of the XML Schema, the XML file can contain just one representation of the Package Data, while having multiple records of the MPN data. This concept eliminates data duplicity, minimizes data file size, and makes the transfer of data highly efficient. The linking of different xml components of data is enabled using ID's.

4.2 Business Info

path	PartModel/BusinessInfo
diagram	
type	JEP30-D10:BusinessInfoType, RequestType, ResponseType, DeclarationType.

The Business Information section contains a structure to capture the details of the person who is requesting the data from the supplier, plus the details of the person and authorizer who is responding to that request.