

How to use CFormat Ver1.2

2019/6/1

<http://www.lpb-forum.com/>

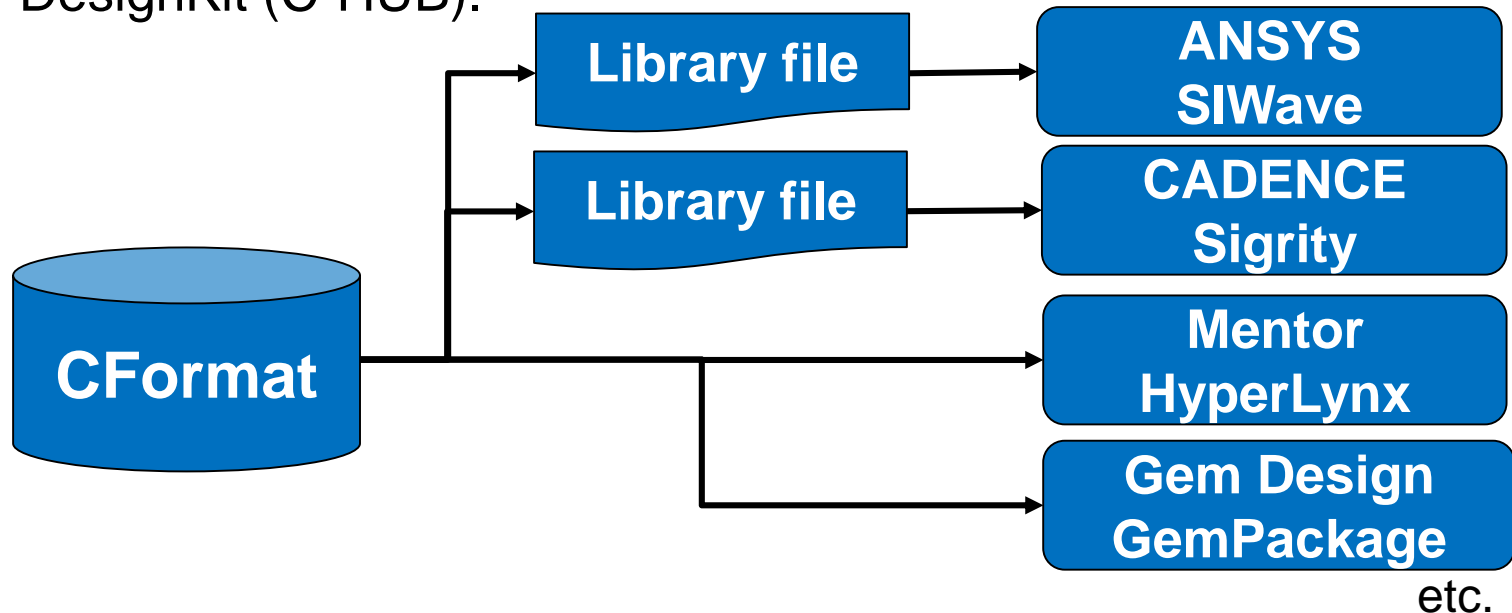
Table of contents.

- [Purpose of IPB Format](#)
- [LPB Format Overview](#)
- [What can I do with LPB format?](#)
- [LPB design kit, How to use “CHUB”](#)
- [Example of simple C-Format](#)

What's New

2019/2/28:

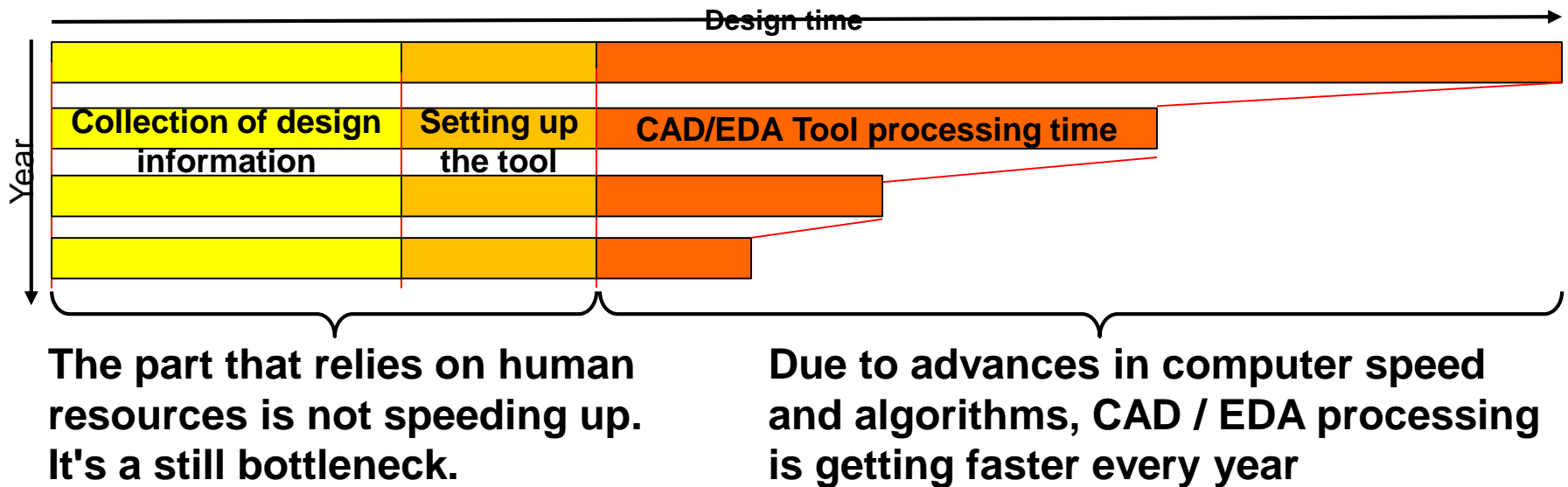
- The CFormat Library for ceramic capacitors has been released by Murata Manufacturing Co., Ltd. .
<https://www.murata.com/ja-jp/tool/c-format>
- EDA tool can read C-Format file directory, and you can convert the CFormat to ANSYS/Siwave, CADENCE/Sigrity libraries using LPB DesignKit (C-HUB).



PURPOSE OF LPB FORMAT

Challenges in the design

Due to advances in computer speed and algorithms, CAD / EDA processing is getting faster every year. However, a period of time spent collecting design information and configuring EDA tools is still a bottleneck as it relies on human resources.



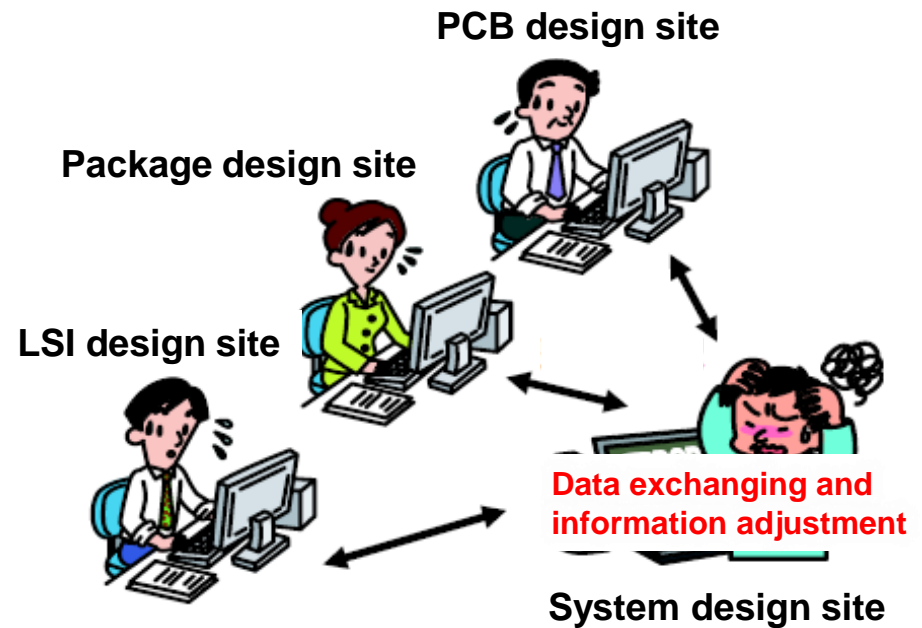
To shorten the development period, it is necessary to improve the efficiency of the part relying on human resources.

Purpose of LPB Format

Recent design styles have shifted to horizontal international specialization. In this style, a system is designed by several design houses having own design culture. We should adjust manually the following differential of design house's culture.

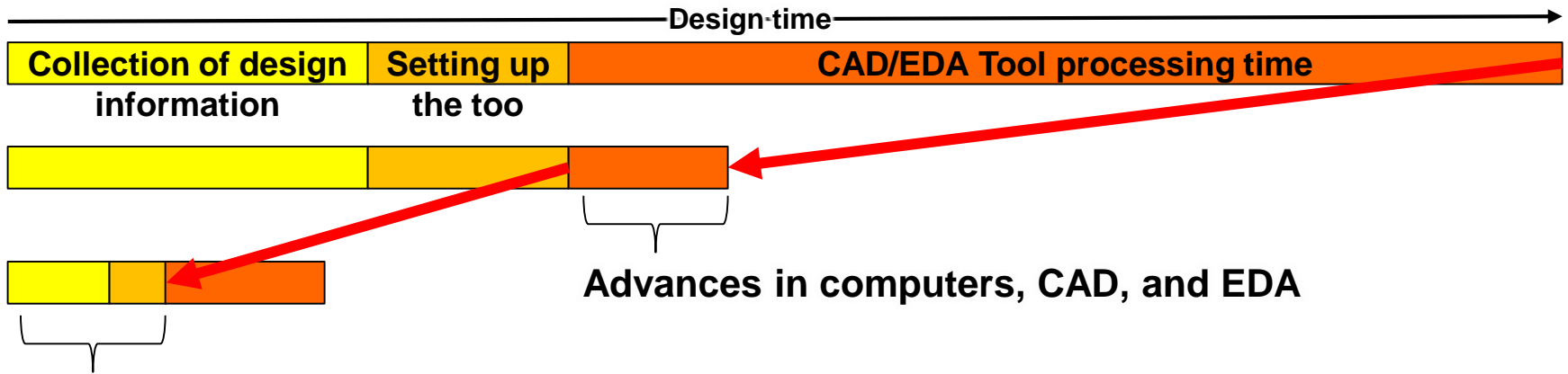
- Difference in form of design specification.
- Difference in file format of EDA tool to use.
- Difference of technical terms

The ratio of such kind of work that dependent on human resources is increasing year by year.

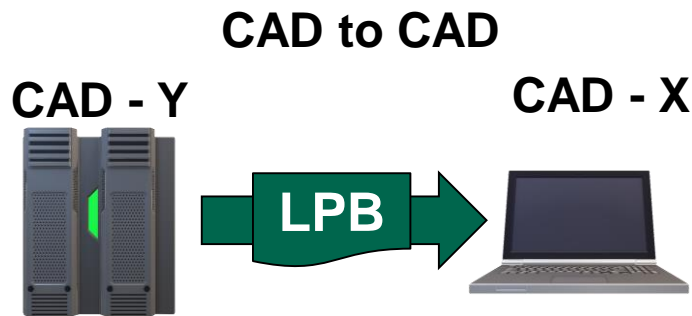


LPB Format is a standard to improve the efficiency of parts relied on human resources.

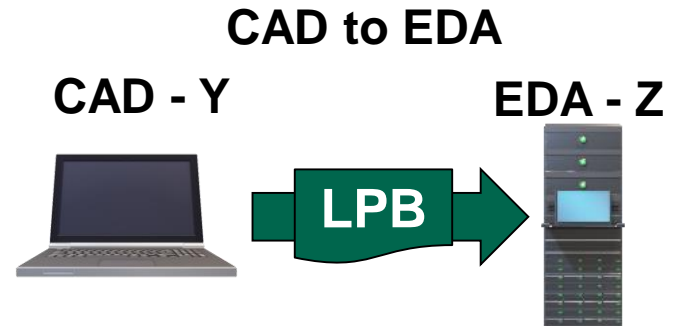
Effects of using LPB Format



LPB Format improves the efficiency of parts relied on human resources.



LPB Format reduces the troubles of data exchange.



LPB Format improves the efficiency of EDA setting.

As for case study, please see the following URL.

http://www.jeita-sdtdc.com/jeita-edatc/wg_lpb/home/docs/eds11nov_sp02.pdf

LPB FORMAT OVERVIEW

What is LPB Format?

LPB Format is a data format to improve the efficiency of work relied on human resources. It is standardized by IEC and IEEE.

Preview

<https://www.techstreet.com/products/preview/1908285>

http://jeita-sdtdc.com/publishedmaterials/lpbformatv3_preview_download/

Sample

http://jeita-sdtdc.com/publishedmaterials/lpbformat_download/

IEC 63055:2016

Format for LSI-Package-Board Interoperable design

<https://webstore.iec.ch/publication/26181>

IEEE2401-2015

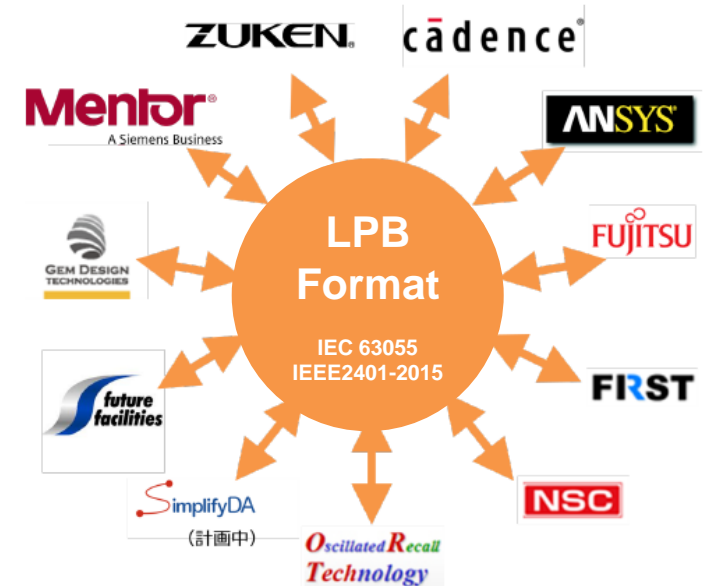
IEEE Standard Format for LSI-Package-Board Interoperable Design

https://www.techstreet.com/standards/ieee-2401-2015?product_id=1908285



LPB Format Adoption Status

- More than 13 CAD/EDA vendors adopte LPB Format as an interface (see left figure).
- Referenced board design data are published by LPB Format
 - Toshiba Electronic Devices & Storage Corporation etc.
- Parts libraries are published byLPB Format.
 - Murata Manufacturing Co., Ltd.
 - Toshiba Electronic Devices & Storage Corporation etc.



LPB Format Configuration

LPB Format consists of the following five data formats.

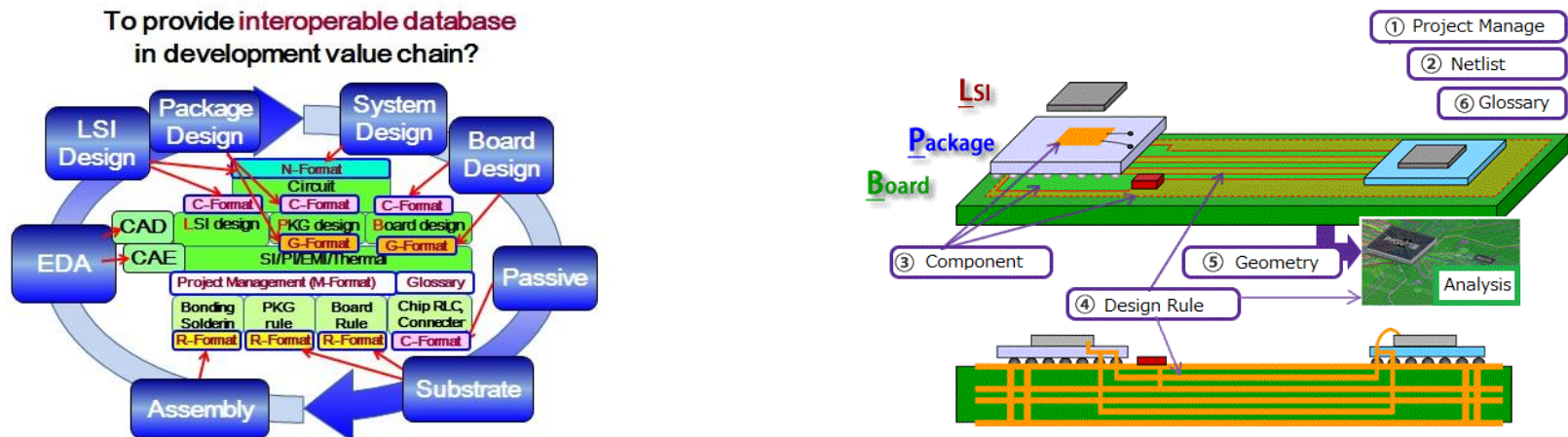
M-Format : Project Management

N-Format : Netlist. Define connection information between parts

C-Format : The component definition. Define design constraints, parts footprints and simulation model.

R-Format : Design rules. Define the layer stack-up and rules such as line & space.

G-Format : Geometry. Graphic data representing layer stack-up structure (2D) such as printed circuit board and IC package



WHAT CAN I DO WITH LPB C-FORMAT?

How do I use C-Format?

Case studies using LPB Format are published in the URL below.

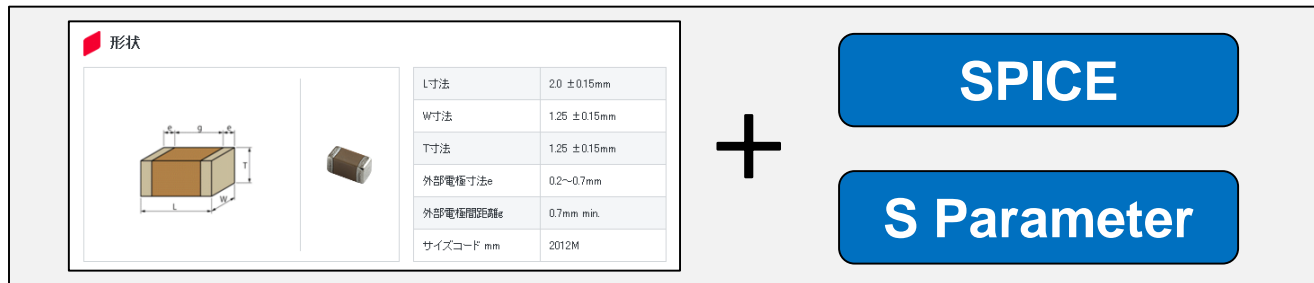
http://jeita-sdtdc.com/publishedmaterials/lpbforum_documents/

This notebook mainly explains the use of CFormat. CFormat contains the following information:

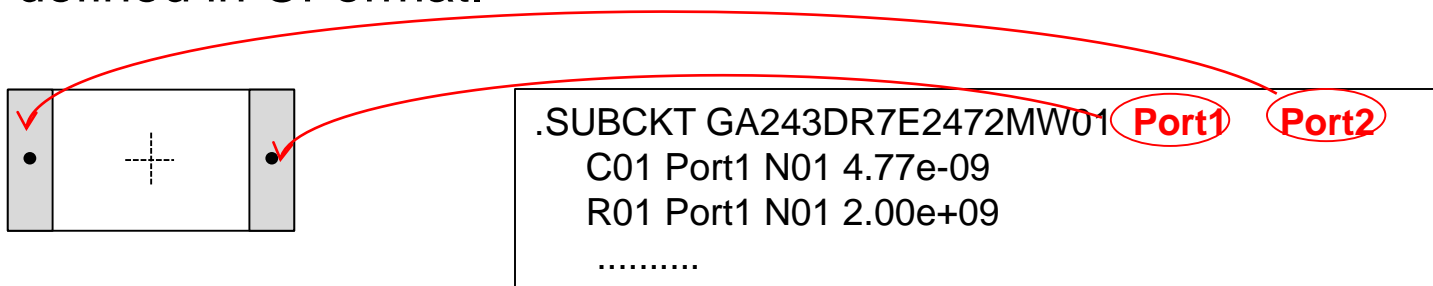
- Geometrical information of parts such as outer shape, foot-print and terminal name etc.
- Cross reference part's terminal with simulation models such as SPICE and IBIS.
- Design constraints such as skew, impedance and delay conditions etc.
- Layout information such as floor plans and part coordinates etc.

Basic form of CFormat

The most basic form of CFormat is a container of foot-print and simulation models.



Cross-reference the part's terminal with the I/O node of simulation model is defined in CFormat.



The CFormat provided by Murata Manufacturing Co., Ltd is also of this form.

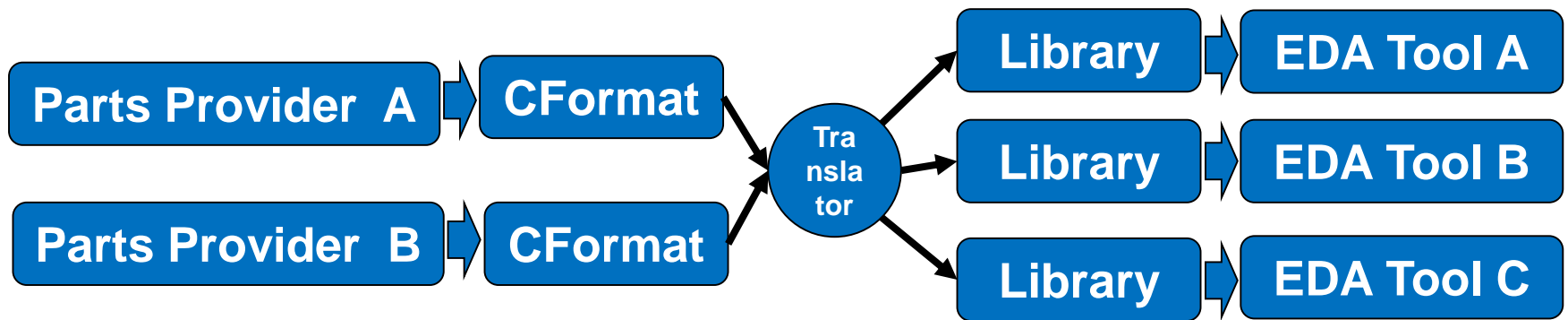
What can I do with CFormat?

Since the physical shape and simulation model have been integrated, the simulation tool can be set up efficiently. We have published the following case studies.

- **PI analysis using LPB format and IBIS5.0**
http://jeita-sdtdc.com/download/lpbforum/20170310_LPB-Forum/6_PI_analysis_with_LPB_Format_and_IBIS5.pdf
- **Reduced turn around time for capacitor optimization using CFormat.**
http://jeita-sdtdc.com/download/lpbforum/20160311_LPB-Forum/6a_DecupingOptimizationByCFoomat.pdf
- **Efficient simulation using Cformat**
http://jeita-sdtdc.com/download/lpbforum/20160311_LPB-Forum/6b_LPBSimulationByScripting.pdf

What can I do with CFormat?

CFormat provided by the part vendor can be converted into libraries of EDA tool using LPB design kit(※). It eliminates the hassle of preparing EDA libraries.



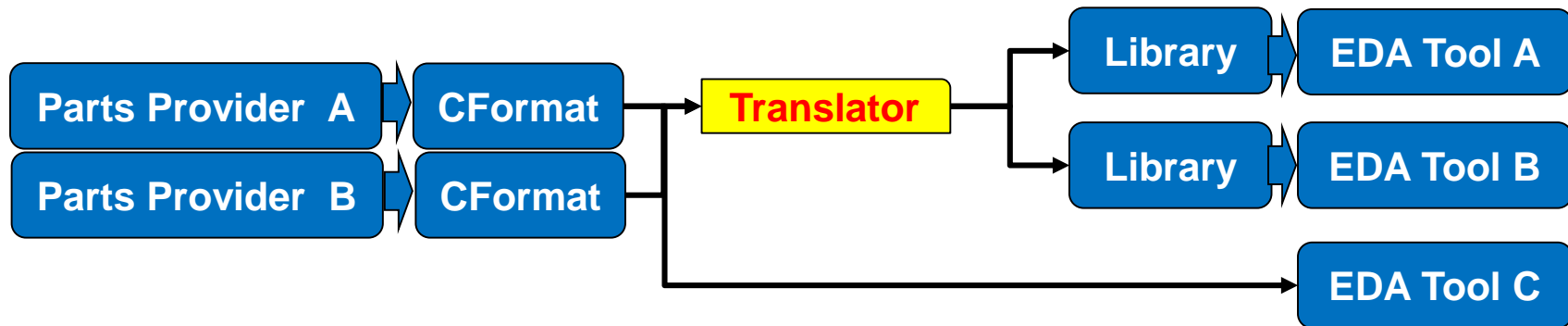
JEITA is encouraging various parts vendors to provide CFormat. At the present (2019/1), CFormat of capacitor component have been provided by Murata Manufacturing Co., Ltd.

(※)The following EDA can import the CFormat directory:

- http://jeita-sdtdc.com/download/lpbforum/20150319_LPB-Forum/3-2_DesignForce_Zuken.pdf
- <http://gemdt.com/gempackage/>

What can I do with CFormat?

Some EDA tools can read CFormat directory for tool setup(※). And CFormat can be converted into libraries of EDA tool using LPB design kit. It eliminates the hassle of preparing EDA libraries.



JEITA is encouraging various parts vendors to provide CFormat. At the present (2019/1), CFormat of capacitor component have been provided by Murata Manufacturing Co., Ltd.

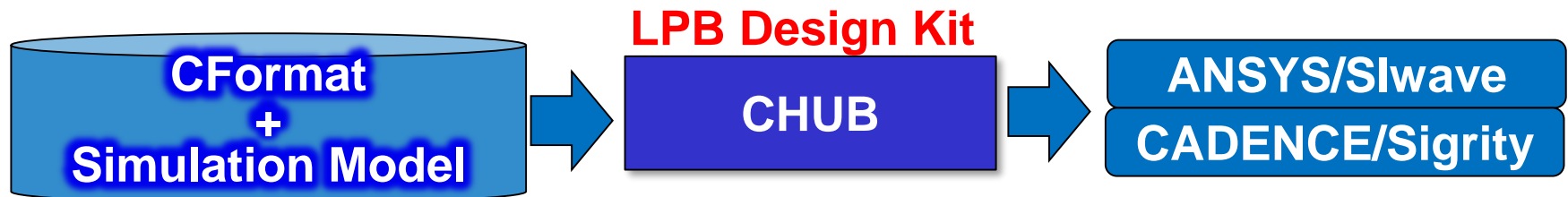
(※)The following EDA can import the CFormat directory:

- Mentor/HyperLynx
- ZUKEN/CR8000
- Gem Design Tech./GemPackage

How do I convert CFormat to EDA library?

The design kit provided by JEITA has a function to convert CFormat (Condenser) into ANSYS / Siwave and CADENCE / Sigrity library. It shows an example of library conversion and BOM management.

<http://jeita-sdtdc.com/lpb-open-source-project/download/>



Since CFormat is defined in XML(※), development of utility tools is easy. The source code is available for the design kit, so please refer to it.

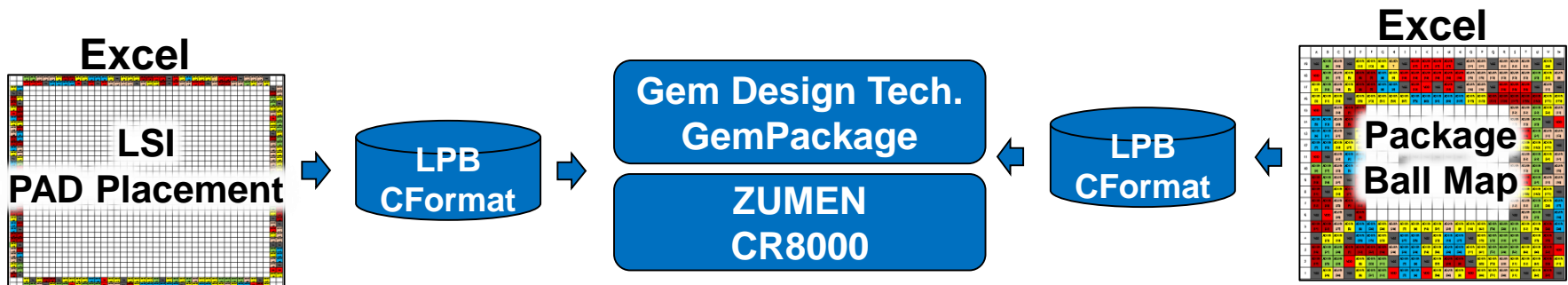
(※) XML : <https://www.google.com/search?q=XML>

Other case study

LPB Format is a royalty-free data format. You can use it to build internal design infrastructure.

For example

Micro soft Excel is often used for design specifications. However, it is difficult to create the interface of EDA tool with Excel file because the form of Excel file is different for each design site. LPB Format can be solved this kind of problem. You can use LPB Format as a central HUB to build tool chain. A simple program is enough to convert Excel to CFormat. The LPB Design Kit (*) includes this kind of reference program.

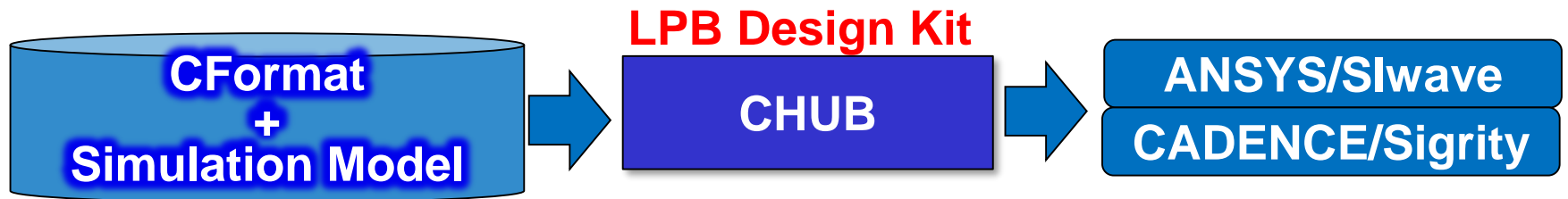


LPB DESIGN KIT

HOW TO USE “CHUB”

What is CHUB ?

CHUB is a tool in LPB DesignKit. It converts CFormat file for capacitor into library files of ANSYS/SIWave and CADENCE/Sigrity.



In additionally it has the following functions.

- Change parts name and family name to internal code name for internal management.
- Change the footprint provided by parts vendor to internal one according to the production rule.

Installation of LPB DesignKit

Please request DesignKit from the following URL.

<http://jeita-sdtec.com/en/lpb-open-source-project/download/>

- **The download URL will be sent back to your email address.**
- **Extract the downloaded zip file into any folder (e.g. LPBDK).**

Installation of LPB CFormat

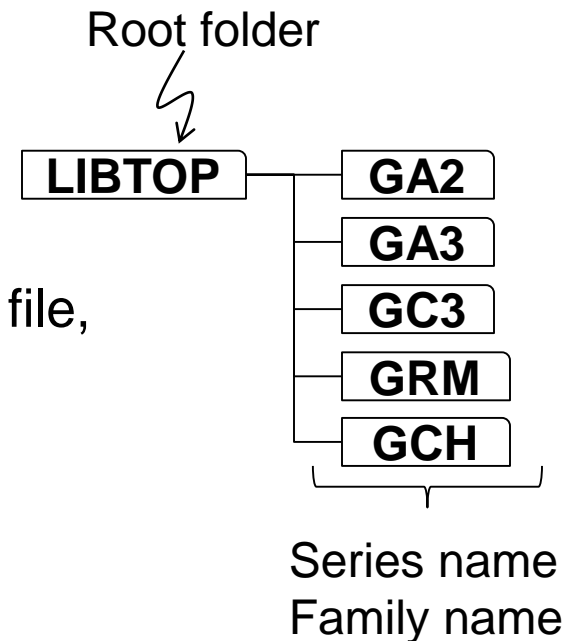
1. The CFormat Library for ceramic capacitors has been released by Murata Manufacturing Co., Ltd. .

<https://www.murata.com/ja-jp/tool/c-format>

CFormat files are packed in ZIP format for each series. Download as many as you need.

2. Extract the downloaded ZIP file into any folder (e.g. LIBTOP as shown on the left figure).

The downloaded ZIP file includes CFormat file, SPICE model file and S-Parameter file.



Launch CHUB

Double-click on cformat_hub.exe (e.g. LPBDK/cformat_hub.exe) to launch CHUB.

Look & feel of CHUB

Buttons to launch command.

Series name area:
(Family name area):

This area shows the name of the folder storing CFormat as the series name (family name).

Message Area

Vendor	InHouse
1 GA2	
2 GA3	
3 GC3	
4 GCD	
5 GCE	
6 GCH	
7 GCJ	
8 GCM	
9 GCQ	
10 GGD	
11 GGM	
12 GI4	
13 GJ8	

Category	Family	File	Module
1 2828_1111	GQM	GQM22M5C2H...	GQM22M5C2H...
2 2828_1111	GQM	GQM22M5C2H...	GQM22M5C2H...
3 2828_1111	GQM	GQM22M5C2H...	GQM22M5C2H...
4 2828_1111	GQM	GQM22M5C2H...	GQM22M5C2H...
5 2828_1111	GQM	GQM22M5C2H...	GQM22M5C2H...
6 2828_1111	GQM	GQM22M5C2H...	GQM22M5C2H...
7 2828_1111	GQM	GQM22M5C2H...	GQM22M5C2H...
8 2828_1111	GQM	GQM22M5C2H...	GQM22M5C2H...
9 2828_1111	GQM	GQM22M5C2H...	GQM22M5C2H...
10 2828_1111	GQM	GQM22M5C2H...	GQM22M5C2H...
11 2828_1111	GQM	GQM22M5C2H...	GQM22M5C2H...
12 2828_1111	GQM	GQM22M5C2H...	GQM22M5C2H...

Messages

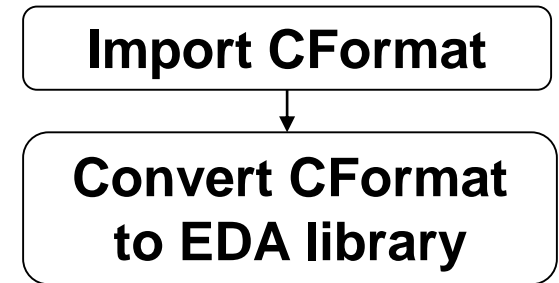
2018-12-25 17:28:18: INFO : 2012_0805
2018-12-25 17:28:27: INFO : 3216_1206
2018-12-25 17:28:32: INFO : 3225_1210
2018-12-25 17:28:33: INFO : 1608_0603
2018-12-25 17:28:56: INFO : 1005_0402

Parts list area:

This area shows the all parts listed in C Format.

Convert CFormat to EDA library

Converting CFormat to EDA library is very simple. It takes only two steps as shown in the flow chart on the left.

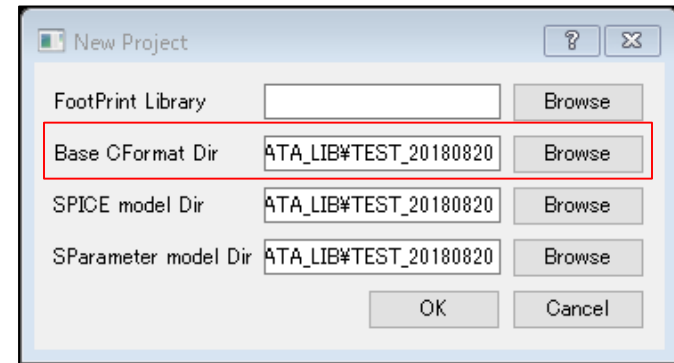
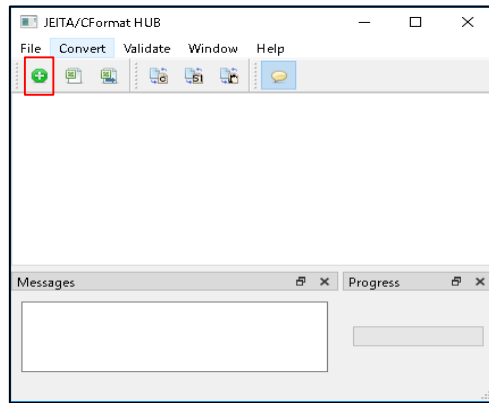


Import CFormat files

File -> New project



- FootPrint Library (optional): Enter the file name of footprint library if you want to replace the footprint contained in the CFormat file with another one.
- Base CFormat Dir: Enter the name of the top folder where the CFormat files are saved (.e.g. LIBTOP).
- SPICE model Dir, SParameter model Dir : Automatically filled in when you enter "Base CFormat Dir".

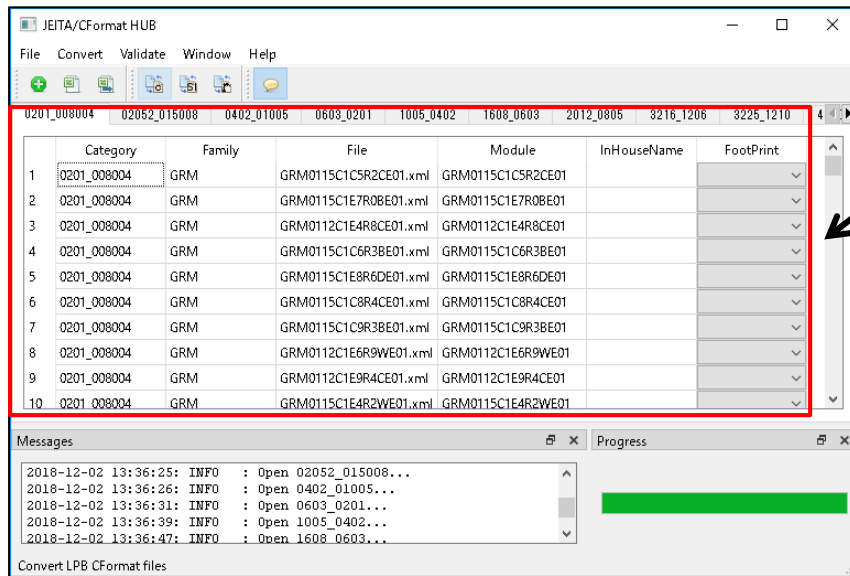


Import CFormat files (cont.)

Press "OK" button to start importing CFormat files.

It will take times to import if the number of files is huge. It took about 20 minutes with Core 2 Duo T7250 (notebook PC about 10 years ago) when import all CFormat files delivered from Murata Manufacturing Co., Ltd. at once.

"Done" dialog will be displayed when finished. Press "YES" to close it.



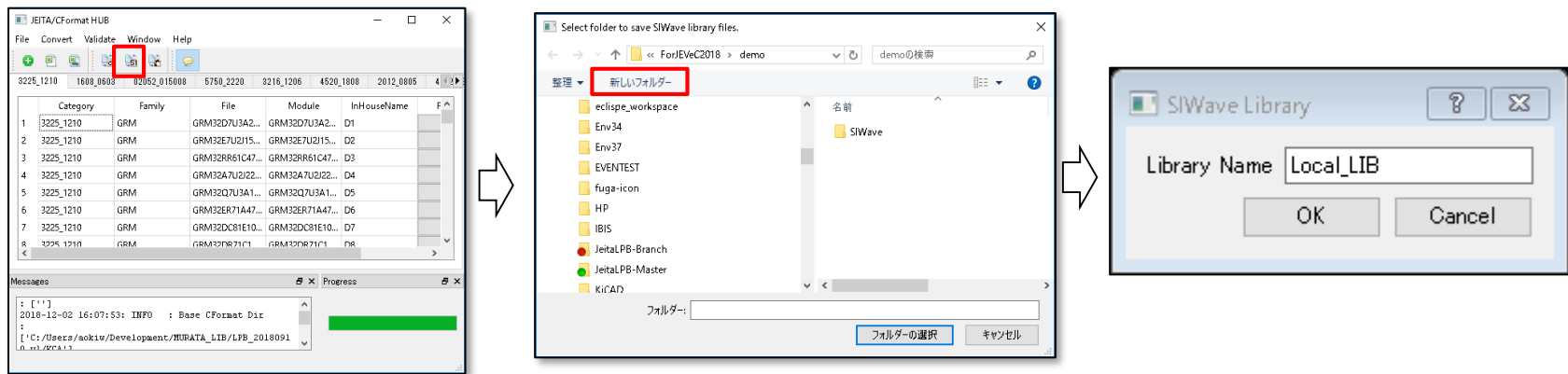
Part list area:
All parts will be listed here
after importing CFormat files.

Convert CFormat to SIwave library

Convert -> SIwave Lib



Select a folder to save the SIwave library. If you do not have the desired folder, press "New Folder" to create a new folder.



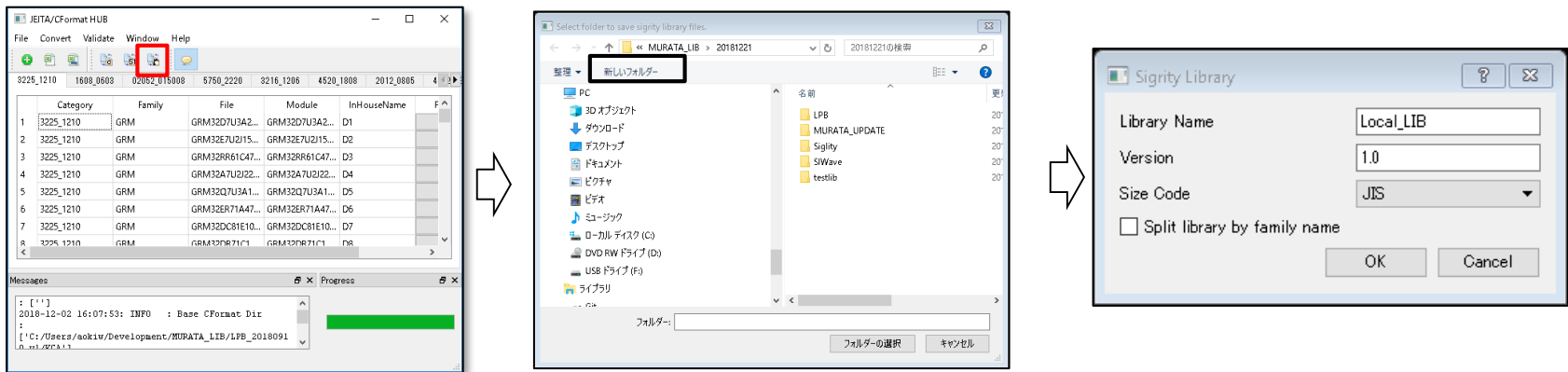
- Library Name: Library name displayed in SIwave. Please enter any name. The default is Local_LIB.

Press OK button to start conversion.

Convert CFormat to Sigrity library


Convert -> Sigrity Lib 

Select a folder to save the Sigrity library. If you do not have the desired folder, press "New Folder" to create a new folder.



- Library Name: Library name displayed in Sigrity. Please enter any name. The default is Local_LIB.
- Version: Version number for library management. Please enter your desired version number.

Convert CFormat to Sigrity library (cont.)

- Size Code: Type  of size code managed by Sigrity. Please select either JIS or EIA.
- Split library by family name: Check this box to separate library files by series. By default all series are managed in one library file.

Press OK button to start conversion.

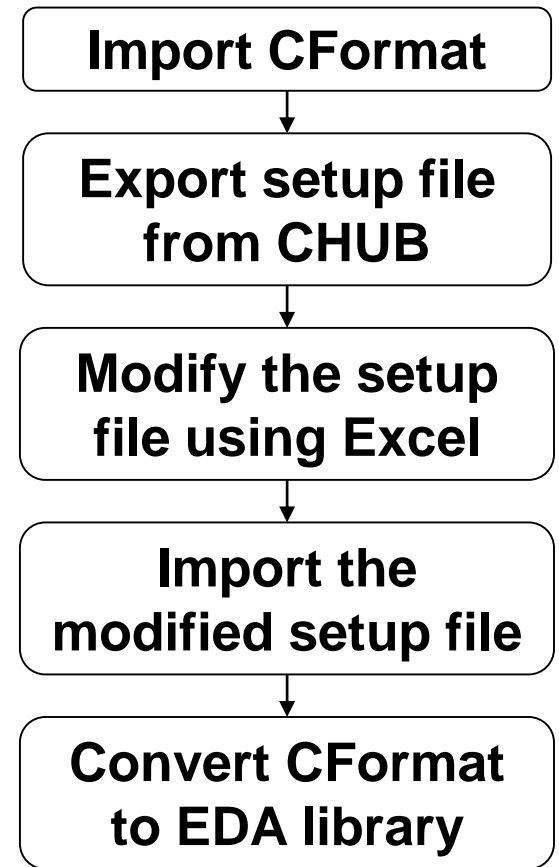
BOM management

In most designs, parts are managed by internal code name that are registered in the BOM. Therefore, need to change the part name to internal code name before entering a model file into EDA tool.

CHUB can replace the part name to internal code name when converting CFormat to EDA tool library.

CHUB exports a template file for tool setting in Excel format. You add internal code name to this setup file and feed it to CHUB.

The flow chart on the right shows the steps to replace the part name to internal code name.



Export setup file from CHUB

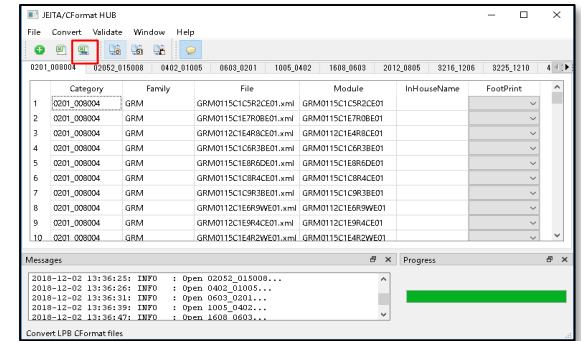
File -> Export Setup file



Export a template file (Excel file) for tool settings. This file contains all the parts listed in the CFormat files.

The template file includes following sheets.

- FamilyName : This sheet shows the name of the folder storing CFormat as the series name (family name).
- Category(size code): Sheets other than FamilyName one are named by size code of SMD. All parts are classified by the size code.



Modify the setup file

FamilyName :

If you want to replace the series name (family name) which provided from part vendor to another one, enter the any name in this sheet.

	A	B	C	D	E	F	G
1	# 2018/12/24 13:46:55						
2	//base_cfor	C:\Users\%aokiw\Development					
3	//footprint_lib						
4	//spice_dir	C:\Users\%aokiw\Development					
5	//spara_dir	C:\Users\%aokiw\Development					
6	#Vendor	#InHouse					
7	GA2						
8	GA3						
9	GC3						
10	GCD						
11	GCE						
12	GCH						
13	GCJ						
14	GCM						
15	GCQ						
16	GGD						
17	GGM						

If you want to replace the series name, enter the any name here. The series name is not changed if the cell is blank.

Modify the setup file (cont.)

Part name :

The sheet named part size contains the parts of that size. If you want to replace the part name to internal code name, enter the internal code name in the #InHouseName column.

#Category	#Family	#File	#Module	#InHouseName	#FootPrint
5750_2220	GRM	GRM55DR72-J224KW01.xml	GRM55DR72-J224KW01		
5750_2220	GRM	GRM55ER72A475KA01.xml	GRM55ER72A475KA01		
5750_2220	GRM	GRM55RB11H105KA01.xml	GRM55RB11H105KA01		
5750_2220	GRM	GRM55DR72H334KW10.xml	GRM55DR72H334KW10		
5750_2220	GRM	GRM55DR73A683KW01.xml	GRM55DR73A683KW01		
5750_2220	GRM	GRM55D7U2H473JW31.xml	GRM55D7U2H473JW31		
5750_2220	GRM	GRM55DR72E334KW01.xml	GRM55DR72E334KW01		
5750_2220	GRM	GRM55RR11H105KA01.xml	GRM55RR11H105KA01		
5750_2220	GRM	GRM55EB11H475KA01.xml	GRM55EB11H475KA01		
5750_2220	GRM	GRM55DR72E105KW01.xml	GRM55DR72E105KW01		
5750_2220	GRM	GRM55D7U3A103JW31.xml	GRM55D7U3A103JW31		
5750_2220	GRM	GRM55DR72J154KW01.xml	GRM55DR72J154KW01		
5750_2220	GRM	GRM55DR61H106KA88.xml	GRM55DR61H106KA88		
5750_2220	GRM	GRM55ER61H475KA01.xml	GRM55ER61H475KA01		
5750_2220	GRM	GRM55RR71H105KA01.xml	GRM55RR71H105KA01		
5750_2220	GRM	GRM55DR11E106KA01.xml	GRM55DR11E106KA01		
5750_2220	GRM	GRM55D7U2J473JW31.xml	GRM55D7U2J473JW31		

Fill internal code name in #InHouseName column if you want to replace the part name.

Import the modified setup file

File -> Import Setup file



Launch CHUB again if you already quit it and import the modified setup file.

The screenshot shows the JEITA/CFormat HUB application window. The main area contains a table with the following columns: Category, Family, File, Module, InHouseName, and FootPrint. The 'InHouseName' column is highlighted with a red box. Below the table is a 'Messages' window showing log output.

Category	Family	File	Module	InHouseName	FootPrint
3225_1210	GRM	GRM32D7U3A2...	GRM32D7U3A2...	D1	
3225_1210	GRM	GRM32E7U2J15...	GRM32E7U2J15...	D2	
3225_1210	GRM	GRM32RR61C47...	GRM32RR61C47...	D3	
3225_1210	GRM	GRM32A7U2J22...	GRM32A7U2J22...	D4	
3225_1210	GRM	GRM32Q7U3A1...	GRM32Q7U3A1...	D5	
3225_1210	GRM	GRM32ER71A47...	GRM32ER71A47...	D6	
3225_1210	GRM	GRM32DC81E10...	GRM32DC81E10...	D7	
3225_1210	GRM	GRM32DR71C1	GRM32DR71C1	D8	

Messages window content:

```
: ['']  
2018-12-02 16:07:53: INFO : Base CFormat Dir  
:  
['C:/Users/aokiw/Development/MURATA_LIB/LPB_2018091  
0.../KCA...']
```

The screenshot shows the JEITA/CFormat HUB application window. The main area contains a table with the following columns: Category, Family, File, Module, InHouseName, and FootPrint. The 'InHouseName' column is highlighted with a red box. Below the table is a 'Messages' window showing log output.

Category	Family	File	Module	InHouseName	FootPrint
0201_00804	GRM	GRM015C15R3DCE01.xml	GRM015C15R3DCE01		
0201_00804	GRM	GRM015C1E7R0BE01.xml	GRM015C1E7R0BE01		
0201_00804	GRM	GRM012C1E4R3CE01.xml	GRM012C1E4R3CE01		
0201_00804	GRM	GRM015C1C6R3BE01.xml	GRM015C1C6R3BE01		
0201_00804	GRM	GRM015C1E6R6CE01.xml	GRM015C1E6R6CE01		
0201_00804	GRM	GRM015C1C8R3CE01.xml	GRM015C1C8R3CE01		
0201_00804	GRM	GRM015C1C9R3BE01.xml	GRM015C1C9R3BE01		
0201_00804	GRM	GRM012C1E6R9WE01.xml	GRM012C1E6R9WE01		
0201_00804	GRM	GRM012C1E6R4CE01.xml	GRM012C1E6R4CE01		
0201_00804	GRM	GRM015C1E4R2WE01.xml	GRM015C1E4R2WE01		

Messages window content:

```
2018-12-02 13:36:25: INFO : Open 02052_015008...  
2018-12-02 13:36:26: INFO : Open 0402_01005...  
2018-12-02 13:36:31: INFO : Open 0603_0201...  
2018-12-02 13:36:39: INFO : Open 1005_0402...  
2018-12-02 13:36:47: INFO : Open 1608_0603...
```

The internal code names listed in the setup files are displayed here.

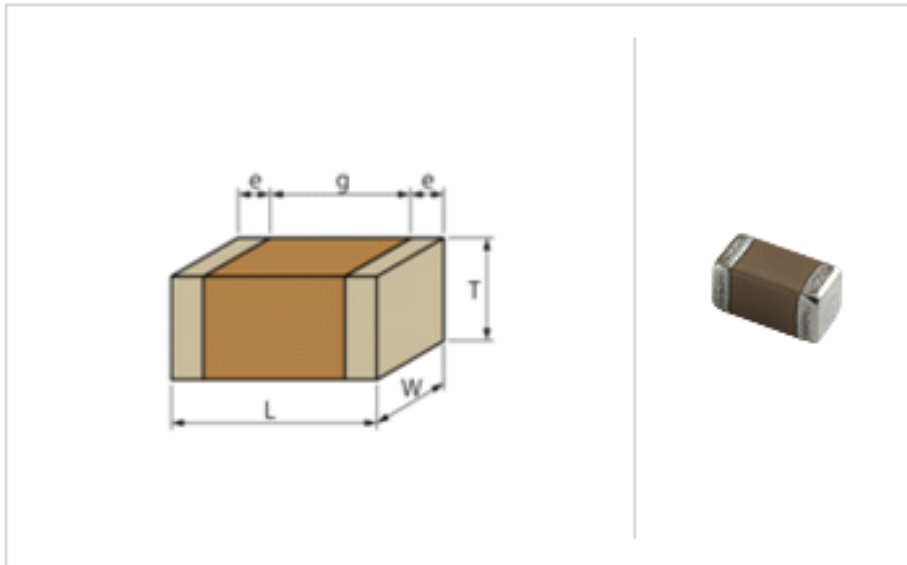
EXAMPLE OF SIMPLE C-FORMAT

Example of a simple CFormat

Let's see a Cformat file delivered from Murata Manufacturing Co., Ltd. .
(please see next page)

□ GRM21BB30J226ME38

形状



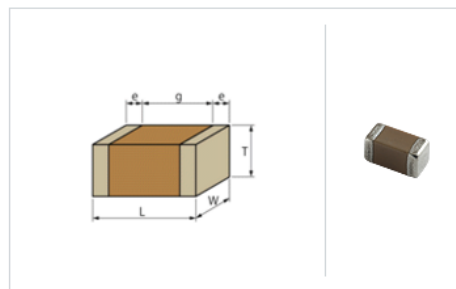
L寸法	2.0 ±0.15mm
W寸法	1.25 ±0.15mm
T寸法	1.25 ±0.15mm
外部電極寸法e	0.2~0.7mm
外部電極間距離g	0.7mm min.
サイズコード mm	2012M

GRM21BB30J226ME38

```
<?xml version="1.0" ?>
<LPB_CFORMAT version="2020">
<header company="MURATA" date="Wednesday Dec. 19 2018"
  design_revision="1.0" project="GRM"/>
<global>
  <unit>
    <distance unit="mm"/>
    <capacitance unit="uF"/>
    <resistance unit="ohm"/>
    <inductance unit="nH"/>
  </unit>
  <shape>
    <rectangle height="1.25" id="1" width="2"/>
    <rectangle height="1.25" id="2" width="0.45"/>
  </shape>
  <padstack_def>
    <padstack id="1">
      <ref_shape shape_id="2" x="0" y="0" pad_layer="BOTTOM"/>
    </padstack>
  </padstack_def>
</global>
<module name="GRM21BB30J226ME38" shape_id="1" thickness="0"
  type="C" x="0" y="0">
  <size_code imperial="0805" metric="2012"/>
  <socket name="socket">
    <default>
      <port_shape padstack_id="1"/>
    </default>
    <port id="1" x="-0.775" y="0.0"/>
    <port id="2" x="0.775" y="0.0"/>
  </socket>
</specification>
  <capacitance typ="22"/>
</specification>
```

```
<reference xmlns:spice="http://www.jeita.or.jp/LPB/spice"
  reffile = "GRM21BB30J226ME38.mod"
  format="SPICE" >
  <connection socket_name="socket" port_id="1">
    <spice:ref_port subckt="GRM21BB30J226ME38" portid="1"/>
  </connection>
  <connection socket_name="socket" port_id="2">
    <spice:ref_port subckt="GRM21BB30J226ME38" portid="2"/>
  </connection>
</reference>
<reference
  xmlns:touchstone="http://www.jeita.or.jp/LPB/touchstone"
  reffile = "GRM21BB30J226ME38.s2p"
  format="TOUCHSTONE" >
  <connection socket_name="socket" port_id="1">
    <touchstone:ref_port portid="1"/>
  </connection>
  <connection socket_name="socket" port_id="2">
    <touchstone:ref_port portid="2"/>
  </connection>
</reference>
</module>
</LPB_CFORMAT>
```

形状



L寸法	20 ±0.15mm
W寸法	1.25 ±0.15mm
T寸法	1.25 ±0.15mm
外部電極寸法e	0.2~0.7mm
外部電極間距離e	0.7mm min.
サイズコード mm	2012M

XML

M-Format, C-Format and R-Format use the World Wide Web Consortium standard for the XML version 1.0.

The element of XML is enclosed in triangular brackets < >.

```
<element_name>
```

An element can contain multiple elements as its children. An element start with `<element_name>` and end with `</element_name>` if the element contains children.

```
<element_name>....</element_name>
```

If an element does not contain children, the element simply expresses as `<element_name/>`.

An element can have one or more attributes. An attribute is represented by name, and value enclosed in double quotation.

```
attribute_name="attribute_value"
```

C-Format configuration

```
<?xml version="1.0" ?>
```

```
<JEITA_LPB_CFORMAT version="2020">
```

```
<header company="MURATA"  
  date="Dec.19 2018"  
  design_revision="1.0"  
  project="GRM"/>
```

<header> element

```
<global>  
  :  
</global>
```

<global> element

```
<module name="GRM21BB30J226ME38"  
  shape_id="1" thickness="0"  
  type="C" x="0" y="0">  
  :  
</model>
```

<module> element

```
</JEITA_LPB_CFORMAT>
```


<header> element

```
<header  
  company="MURATA"  
  date="Wednesday Dec. 19 2018"  
  design_revision="1.0"  
  project="GRM"  
>
```

project

Specifies the name of the project (required).

design_revision

Specifies the revision number for the design (required).

company

Specifies the name of organization delivering this CFormat file (optional).

data

Delivered date (optional).

<global> element

```
<global>
```

```
<unit>
```

```
:
```

```
</unit>
```

<unit> element

```
<shape>
```

```
:
```

```
</shape>
```

<shape> element

```
<padstack_def>
```

```
:
```

```
</padstack def>
```

<padstack_def> element

```
</global>
```

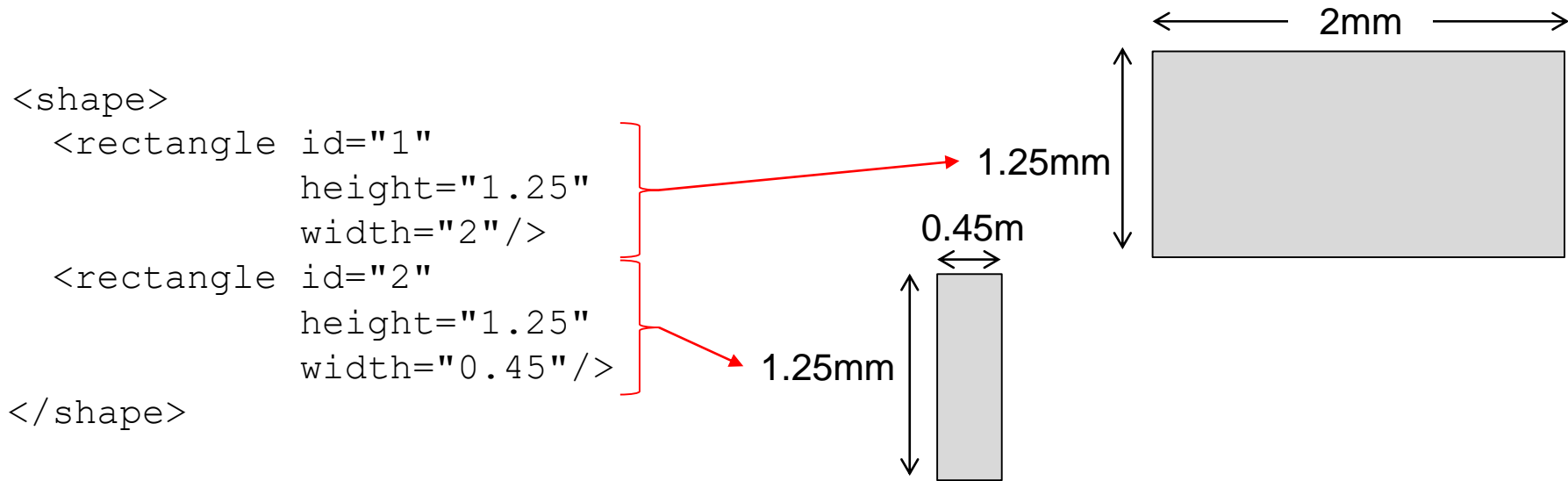
The <global> element defines the unit system, basic shapes and padstack to be used throughout the CFormat files. The scope of the defined variables is limited to the file in which it is declared. The content of the <global> element consists of one <unit> element and one or zero <shape> and <padstack_def> elements.

<unit> element

```
<unit>  
  <distance unit="mm"/>  
  <capacitance unit="uF"/>  
  <resistance unit="ohm"/>  
  <inductance unit="nH"/>  
</unit>
```

The <unit> element defines the unit system to be used in CFormat.
Left example defines the unit system of length, capacitance, resistance and inductance.

<shape> element



The <shape> element defines basic shapes those are referenced by other attributes and elements, such as a <padstack_def> element.

id

Specifies the unique identifier that is used to reference the shape from other attributes and elements.

width

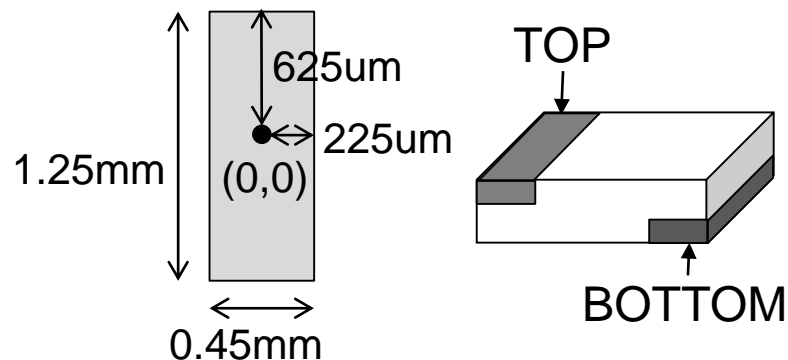
Specifies the width of the rectangle.

height

Specifies the height of the rectangle.

<padstack_def> element

```
<padstack_def>
  <padstack id="1">
    <ref_shape shape_id="2"
      x="0" y="0"
      pad_layer="BOTTOM"/>
  </padstack>
</padstack_def>
```



The <padstack_def> element use to define pad stacks. Via and pin are defined by pad stacks.

shape_id :

Specifies the identifier of the predefined shape that is defined at the <shape> element.

x, y :

Specifies the location of the reference point of the shape with respect to the local origin.

pad_layer :

Specifies the placement side of the shape. In case of pin of SMD, it should be BOTTOM.

<module> element

```
<module name="GRM21BB30J226ME38"  
  shape_id="1"  
  thickness="0"  
  type="C"  
  x="0" y="0">
```

```
<size_code/>
```

<size_code> element

```
<socket>
```

```
:
```

<socket> element

```
</socket>
```

```
<specification>
```

```
:
```

<specification> element

```
</specification>
```

```
<reference>
```

```
:
```

<reference> element

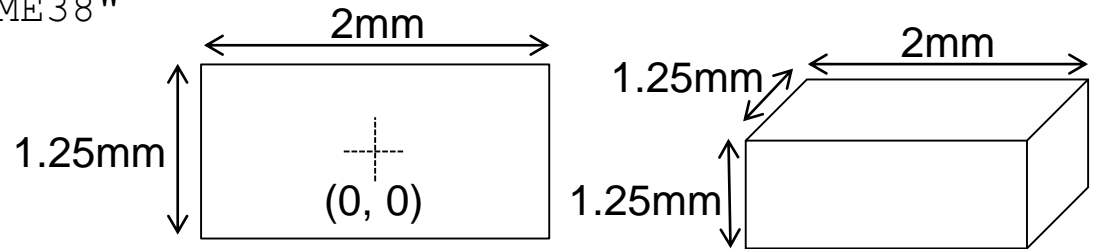
```
</reference>
```

```
</module>
```

The module is the basic design unit of items such as a die or LSI package. The <module> element encapsulates the geometry information, design constraints, I/O interface, and electrical model of a module.

<module> element (cont.)

```
<module name="GRM21BB30J226ME38"  
  shape_id="1"  
  thickness="1.25"  
  type="C"  
  x="0" y="0">
```



name

Specifies the module name or part name.

type

Specifies the module type. C means capacitor.t.

shape_id

Specifies the identifier of the predefined shape to define the boundary shape of the module. The above example refers the shape of id=1.

x, y

Specify the location of the reference point of the shape with respect to the local origin.

thickness

Specifies the thickness of module.

<size_code> element

```
<size_code imperial="0805"  
            metric="2012"/>
```

The <size_code> element defines the size of SMD. It's given in imperial or metric units.

`metric`

Specifies the size code given metric unit (JIS).

`imperial`

Specifies the size code given imperial unit (EIA).

<socket> element

```
<socket name="socket">
```

```
<default>
```

```
<port_shape padstack_id="1"/>
```

<default> element

```
</default>
```

```
<port id="1" x="-0.775" y="0.0"/>
```

<port> element

```
<port id="2" x="0.775" y="0.0"/>
```

```
</socket>
```

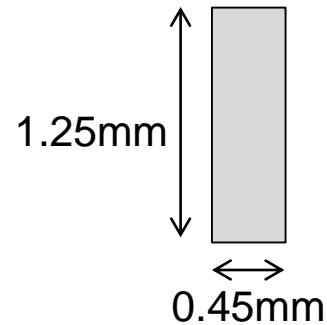
The <socket> element defines the I/O ports. The port definition includes not only geometrical information but also logical information. Above example includes only geometrical information.

name

Specifies the socket name that is used to reference the socket from other attributes and elements.

<default> element

```
<socket name="socket">  
  <default>  
    <port_shape padstack_id="1"/>  
  </default>  
  <port id="1" x="-0.775" y="0.0"/>  
  <port id="2" x="0.775" y="0.0"/>  
</socket>
```



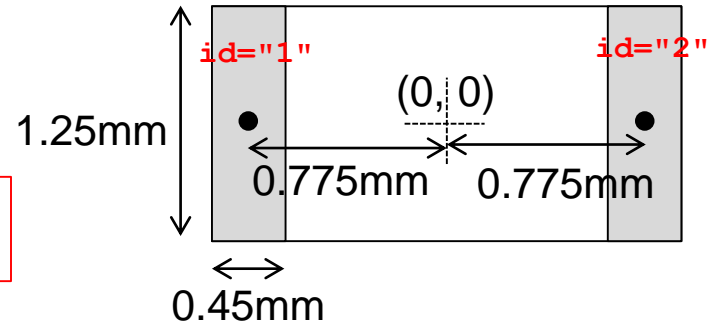
The <default> element is used to define the shape of ports if shape of all ports are same. If each ports have different shape, the shape is defined in <port> element.

padstack_id

Specifies the identifier of the predefined padstack that is used to define the default pad shape. Above example referes the padstack of id=1.

<port> element

```
<socket name="socket">
  <default>
    <port_shape padstack_id="1"/>
  </default>
  <port id="1" x="-0.775" y="0.0"/>
  <port id="2" x="0.775" y="0.0"/>
</socket>
```



The <port> element defines the logical and geometry information for a port of the module. In above example, only geometry information (shape and location) is defined. More over , shape of port is defined by <default> element cause of all ports have same shape.

id

Specifies the unique identifier that is used to reference the port from other attributes and elements.

x, y

Specify the location of the reference point of the referenced padstack with respect to the local origin of the module.

<specification> element

```
<specification>  
  <capacitance typ="12.5"/>  
</specification>
```

The <specification> element defines the specifications of the module itself, such as power consumption, resistance or capacitance etc. In this example, only capacity value (typical value) are defined.

<reference> element

```
<reference
  xmlns:spice="http://www.jeita.or.jp/LPB/spice"
  format="SPICE"
  reffile = "GRM21BB30J226ME38.mod">
```

SPICE

```
<connection>
  :
</connection>
```

<connection> element

```
</reference>
```

```
<reference
  xmlns:touchstone="http://www.jeita.or.jp/LPB/touchstone"
  format="TOUCHSTONE"
  reffile = "GRM21BB30J226ME38.s2p">
```

S-Parameter

```
<connection>
  :
</connection>
```

<connection> element

```
</reference>
```

The <reference> element is used to make a relationship between the module and electrical, thermal or geometrical model file. . In this example, SPICE and S-Parameter (or TOUCHSTONE) model are related with the module.

<reference> element (cont.)

```
<reference
  xmlns:spice="http://www.jeita.or.jp/LPB/spice"
  format="SPICE"
  reffile = "GRM21BB30J226ME38.mod">
  :
</reference>
```

SPICE

```
<reference
  xmlns:touchstone="http://www.jeita.or.jp/LPB/touchstone"
  format="TOUCHSTONE"
  reffile = "GRM21BB30J226ME38.s2p">
  :
</reference>
```

S-Parameter

xmlns:

The namespace of XML are fixed by referenced model.

format

Specifies the model type of reference file.

reffile

Specifies the name of a file with which to make a relationship.

<connection> element

```
<connection socket_name="socket"  
            port_id="1">  
:  
</connection>
```

The <connection> element defines the relationship between a <port> that is defined in the <socket> element and the I/O node of a reference model file.

socket_name

Specifies the name of a socket .

port_id

Specifies the identifier of a port.

The above example refers <port> of id="1" which included in <socket> named "socket".

```
<socket name="socket">  
  <port id="1" ..... />
```

<connection> for SPICE model

```
<connection socket_name="socket" port_id="1">  
  <spice:ref_port subckt="GRM21BB30J226ME38" portid="1"/>  
</connection>  
<connection socket_name="socket" port_id="2">  
  <spice:ref_port subckt="GRM21BB30J226ME38" portid="2"/>  
</connection>
```

In the case of SPICE, a relationship is created by the combination of the sub-circuit name and order of I/O node description.

subckt

Specifies the name of a sub-circuit (.subckt) in the reference SPICE file.

portid

Specifies the order of I/O nodes in the .subckt line.
The value shall be an integer of 1 or more.

<reference> for SPICE model

```
<module name="GRM21BB30J226ME38" ... >
<socket name="socket">
  <port id="1" x="-0.775" y="0.0"/>
  <port id="2" x="0.775" y="0.0"/>
</socket>
<reference
  xmlns:spice="http://www.jeita.or.jp/LPB/spice"
  format="SPICE"
  reffile = "GRM21BB30J226ME38.mod" >
  <connection socket_name="socket" port_id="1">
    <spice:ref_port subckt="GRM21BB30J226ME38" portid="1"/>
  </connection>
  <connection socket_name="socket" port_id="2">
    <spice:ref_port subckt="GRM21BB30J226ME38" portid="2"/>
  </connection>
</reference>
```

The first node of subckt "GRM21BB30J226EM38" is related with the port of id="1". And 2nd node is related with port of id="2".

<connection> for S-Parameter model

```
<connection socket_name="socket" port_id="1">  
  <touchstone:ref_port portid="1"/>  
</connection>  
<connection socket_name="socket" port_id="2">  
  <touchstone:ref_port portid="2"/>  
</connection>
```

In the case of S-Parameter, a relationship is created by the order of I/O node description.

portid

Specifies the order of I/O nodes in S-Parameter file (TOUCHSTONE).

The value shall be an integer of 1 or more.

<reference> for S-Parameter model

```
<module name="GRM21BB30J226ME38" ... >
<socket name="socket">
  <port id="1" x="-0.775" y="0.0"/>
  <port id="2" x="0.775" y="0.0"/>
</socket>
<reference
  xmlns:touchstone="http://www.jeita.or.jp/LPB/touchstone"
  reffile = "GRM21BB30J226ME38.s2p"
  format="TOUCHSTONE" >
  <connection socket_name="socket" port_id="1">
    <touchstone:ref_port portid="1"/>
  </connection>
  <connection socket_name="socket" port_id="2">
    <touchstone:ref_port portid="2"/>
  </connection>
</reference>
```

The first node S-Parameter model is related with the port of id="1".
And 2nd node is related with port of id="2".