



LCDB-Pmaster Conversion

Contents

Introduction *Introduction-1*

1. PMASTER (.pcp) to LCDB Conversion

..... 1-1

- *PMASTER (.pcp) to LCDB Conversion* 1-1
- *About Conversion Rules* 1-3
 - * *About User Definition Properties* 1-8
 - * *Converting Symbol Names* 1-9
 - * *Converting Component Numbers (PCMAC_NO)* 1-10
 - * *Displaying Warning / Error Messages* 1-11
 - * *Starting from Shell (Only for UNIX version)* 1-11
- *Precautions* 1-12
 - * *Pin ID Consistency Check* 1-14

2. LCDB to PMASTER (PMA)

..... 2-1

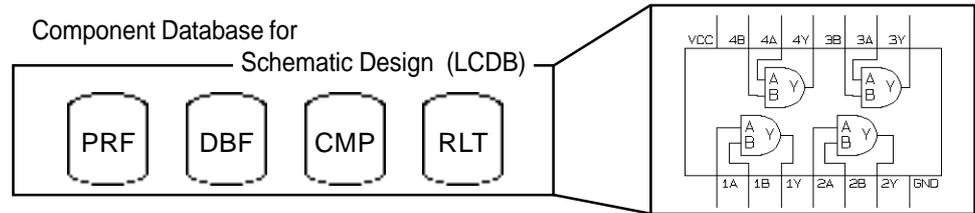
- *LCDB to Component Property Definition File Conversion (.pma)* 2-1
- *About Conversion Rules* 2-3
 - * *About User Definition Properties* 2-7
 - * *Converting Component Number (PCMAC_NO)* 2-8
 - * *Warning / Error Output* 2-9
 - * *Starting from Shell (Only UNIX version)* 2-9
 - * *Converting LCDB to PMASTER* 2-9
- *Precautions* 2-10

Introduction

This Special Training, <System Designer <=> PWS link> explains a procedure for converting a library of System Designer into that of PWS/SWS, its conversion rules, and precautions.

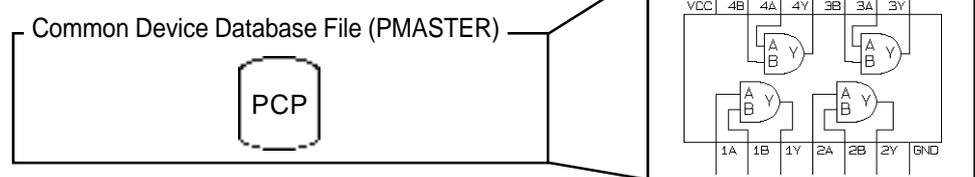
In System Designer, LCDB is provided as a library for defining information on parts.

System Designer



Also in PWS/SWS, PMASTER is provided as a library for defining parts information.

PWS



A similar type of information is registered in LCDB as well as in PMASTER. Therefore, if either one of the files is created, you can also create the other file by using a conversion program.



However, some conversion may not be available depending on the version of System Designer and PWS. Conversion compatibility for available versions is as follows:

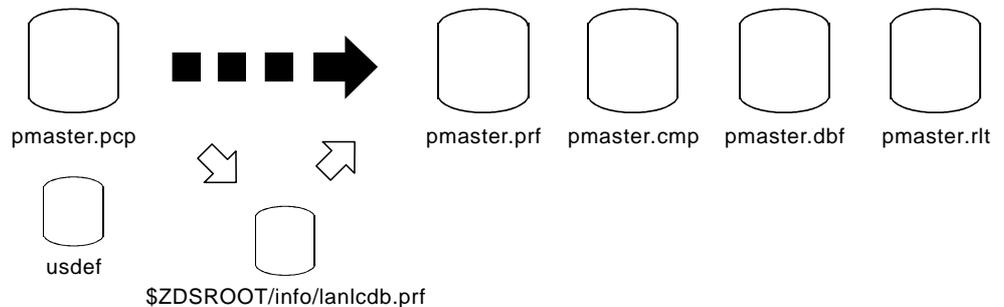
SystemDesigner	PWS	Rev 10.3 or before	Rev 11.0 or later
Rev 3.0 or before	LCDB -> PMA conversion	○	△
	PMaster -> LCDB conversion	○	×
Rev 3.1 or later	LCDB -> PMA conversion	△	○
	PMaster -> LCDB conversion	×	○

1. PMASTER (.pcp) to LCDB Conversion

● PMASTER (.pcp) to LCDB Conversion

Converts PMASTER (.pcp) used for PWS/SWS System into CR-5000 LCDB.

CAUTION To use converted data in System Designer, you need to enter missing information such as a symbol file name.

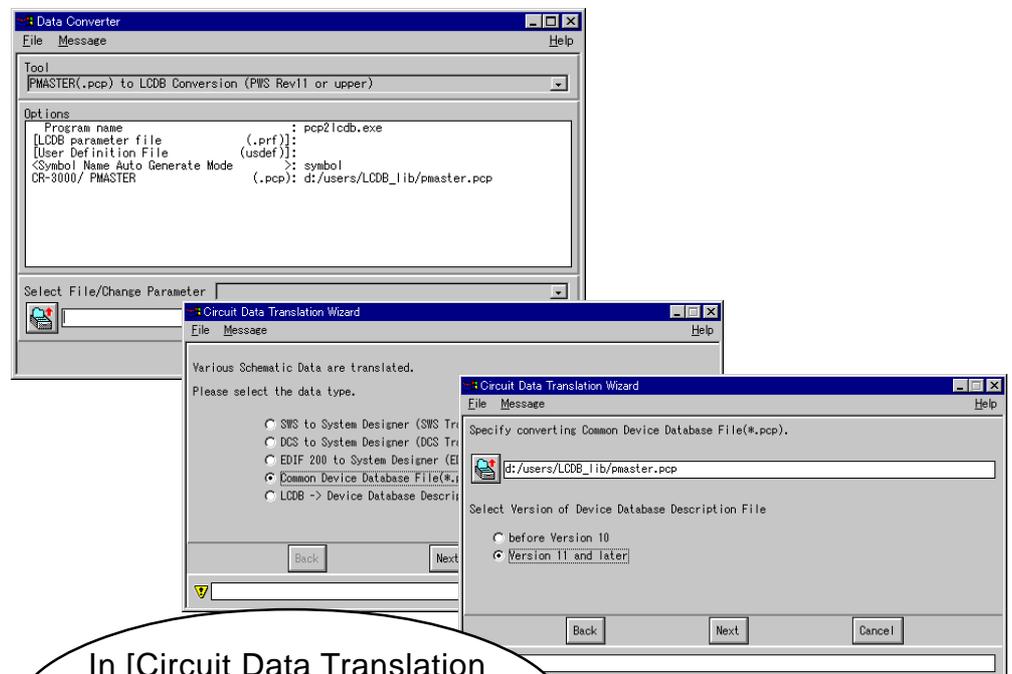


Perform the conversion process for every PMASTER (.pcp).

From the option list of “Tool” in [Data Converter]:

Select [PMMASTER (.pcp) to LCDB Conversion (PWS Rev.11 or upper)] if you are using PWS Rev 11 or later;

Select [PMMASTER (.pcp) to LCDB Conversion (PWS Rev.10 or older)] if you are using PWS Rev 10 or before.



In [Circuit Data Translation Wizard], select [Common Device Database File (.pcp) -> LCDB].



For information on a common device database file, see "PWS Design Preparation Files - Device Database Description File Reference File(.pma)".

Specify the following startup options. (Some of the options are optional.)

Options

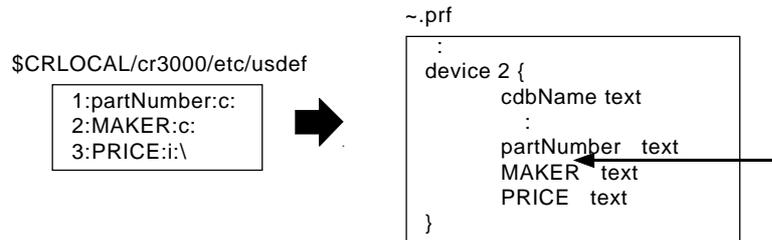
[Program name]: Can not be modified
 (pcp2lcdb.sh) ... For PWS Rev 11 or later
 (pcp2lcdb.exe)
 (pcp2lcdb10.sh) ... For PWS Rev 10 or before
 (pcp2lcdb10.exe)

[LCDB Parameter File (.prf)]: Optional

This option specifies a file name of LCDB to be converted. If the specified file already exists, the parts in the PMASTER are added to the file. However, if a part with the same name exist, information in the existing name overrides the new one. By default, \$ZDSROOT/info/lanlcdb.prf is copied in the same name as that of the Common Device Database File and LCDB is auto-generated.

[User definition file (usdef)]: Optional

This option is specified when information is extracted from user-defined items and transferred to LCDB. If the information is to be transferred to the existing LCDB, a property item name should be described in the LCDB Parameter File (.prf).



If LCDB is auto-generated at the time of conversion, contents of user-defined items will also be extracted automatically.

[Symbol Name Auto Generate Mode] ... Default: OFF

This option specifies whether or not a symbol file name is automatically generated. If you click on the [Symbol Name Auto Generate Mode] line of the Options and set to "symbol", a description of "Symbol name (function name).smb" will be auto-generated.

[Common Device Database File (.pcp)]: Required

This option specifies a Common Device Database File (.pcp) subjected to conversion.

After specifying the startup option, press the <<Execute>> button.

Four files generated by conversion are as follows:

Parameter File	-> pmaster name.prf
Component File	-> pmaster name.cmp
Database File	-> pmaster name.dbf
Component search file	-> pmaster name.rlt

About Conversion Rules

Convert the Common Device Database File (.pcp) into CR-5000 LCDB. Perform the conversion process for every PMASTER (.pcp).

1 gate in 1 package component (without POWER, GROUND pin)



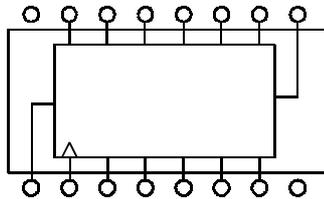
```
$PMA{
  NAME 0.3U : DISCRETE ::1:1:2;
  PCMAC_NO 5102 : 6102 ::;
  SYMBOL{
    0.3U : 1 : Q1,Q2 : Q1=Q2 ;
  }
}
```

Create two components

- Gate (part name_symbol name)
- Package (part name_package)

0.1U_0.1U
0.1U_package

1 gate in 1 package component (with POWER, GROUND pin)



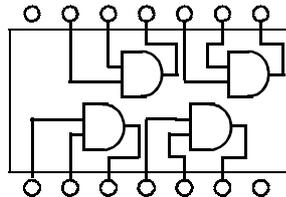
```
$PMA{
  NAME SN74LS165:MODULE:ELSE:1:1:16;
  SYMBOL{
    SN74LS165 : 1 :SL,CK,E,F,G,H,
    %QH%,QH,SI,A,B,C,D,CI ::;
  }
}
```

Create three components

- Gate (part name_symbol name)
- Package (part name_package)
- Power Box (part name_pbox)

0.1U_0.1U
0.1U_package

Multiple gates of the same type in 1 package component (with POWER, GROUND pin)

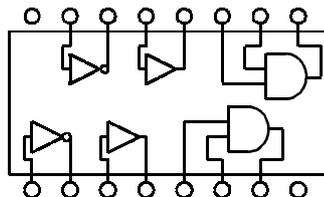


```
$PMA{
  NAME SN74LS08 : MODULE:TTL:1:4:14;
  PCMAC_NO 5014 : 6014 ::;
  SYMBOL{
    2AND :1,2,3,4: A,B,Y : A=B ;
  }
}
```

Create three components

- Gate (part name_symbol name)
- Package (part name_package)
- Power Box (part name_pbox)

Different types of gates in 1 package component (with POWER, GROUND pin)



```
$PMA{
  NAME SN74LS31 : MODULE:TTL:3:6:16;
  PCMAC_NO 5016 : 6016 ::;
  SYMBOL{
    INV : 1,6 : A , Y ::;
    BUF : 2,5 : A , Y ::;
    2NAND : 3,4 : A,B,Y :A=B;
  }
}
```

Create Number of available gate types + two components

- Gate (part name_symbol name)
- Package (part name_package)
- Power Box (part name.pbox)



Rules for creating gates (Component Type: gate)

Part name -> cdbName (ID-name)
partName

Logic = conversion disabled

Part level = conversion disabled

Component number

Symbol information -> Function name (symbol file name)

External pin number -> pinNumber

Pin name -> pinLabel

```

$PMA {
NAME SN74LS08 : MODULE : TTL : 1 : 4 : 14 ;
PCMAC_NO 5014:6014::;
SYMBOL {
    AND2 : 1, 2, 3, 4 : A, B, Y : A = B ;
}
PIN {
1 : 1A : 1 : A
2 : 1B : 1 : B
3 : 1Y : 1 : Y
4 : 2A : 2 : A
5 : 2B : 2 : B
6 : 2Y : 2 : Y
7 : GND : GROUND : GND
8 : 3Y : 3 : Y
9 : 3A : 3 : A
10 : 3B : 3 : B
11 : 4Y : 4 : Y
12 : 4A : 4 : A
13 : 4B : 4 : B
14 : +5V : POWER : +5V
}
CURRENT 40.0
}
    
```

Maximum current consumption = conversion disabled

POWER, GROUND -> io property (VCC, GND)



LCDB Editor

d:/users/LCDB_lib/pmaster.prf

CDB Name	Component Name	Component Name	Component Name
cdbName	componentName	componentName	componentName
SN74LS08	SN74LS08_2AND	SN74LS08_package	SN74LS08.pbo
SN74LS08N	SN74LS08N_SN74LS08N	SN74LS08N_package	SN74LS08N.pbc
SN74LS13	SN74LS13_SN74LS13	SN74LS13_package	SN74LS13.pbo
SN74LS138	SN74LS138_SN74LS138	SN74LS138_package	SN74LS138.pbc
SN74LS138N	SN74LS138N_SN74LS138N	SN74LS138N_package	SN74LS138N.pl
SN74LS139	SN74LS139_SN74LS139	SN74LS139_package	SN74LS139.pbc
SN74LS139N	SN74LS139N_SN74LS139N	SN74LS139N_package	SN74LS139N.pl

Total Count: 92 Search Count: 92

Description of component: part name_symbol name

Edit Component

Component Name: SN74LS08_2AND

Component Type: Gate

Gate Count: 4

Equivalence: 1=2

Block File Name: []

Function Name: 2AND

Symbol File Assignment

Positive Symbol	Negative Symbol
2AND.emb	.

Symbol Pin ID	Common Terminal? isCommonTerminal	Pin Number pinNumber	Pin Label pinLabel	IO io
1	NO	1,4,8,12	A	--
2	NO	2,5,10,13	B	--
3	NO	3,6,8,11	Y	--
4	YES	7	GND	GND
5	YES	14	VCC	VCC

Buttons: Insert Pin, Append Pin, Delete Pin, OK, Cancel



Rules for creating component packages (Component Type: part)

Part name -> cdbName (ID-name) Logic = conversion disable
 partName Part level = conversion disable

Component number Equivalence
 Symbol information -> Function name (symbol file name)

```

$PMA {
  NAME SN74LS08 : MODULE : TTL : 1 : 4 : 14 ;
  PCMAC_NO 5014:6014::;
  SYMBOL {
    AND2 : 1, 2, 3, 4 : A, B, Y : A = B ;
  }
  PIN {
    1 : 1A : 1 : A : ;
    2 : 1B : 1 : B : ;
    3 : 1Y : 1 : Y : ;
    4 : 2A : 2 : A : ;
    5 : 2B : 2 : B : ;
    6 : 2Y : 2 : Y : ;
    7 : GND : GROUND : GND : ;
    8 : 3Y : 3 : Y : ;
    9 : 3A : 3 : A : ;
    10 : 3B : 3 : B : ;
    11 : 4Y : 4 : Y : ;
    12 : 4A : 4 : A : ;
    13 : 4B : 4 : B : ;
    14 : +5V : POWER : +5V : ;
  }
  CURRENT 40.0
}
    
```

External pin number -> pinNumber

External pin name -> pinLabel POWER, GROUND -> io property (VCC, GND)



LCDB Editor

d:/users/LCDB_lib/pmaster.prf

CDB Name	Component Name	Component Name	Component Name
cdbName	componentName	componentName	componentName
SN74LS08	SN74LS08_2AND	SN74LS08_package	SN74LS08.pbo
SN74LS08N	SN74LS08N_SN74LS08N	SN74LS08N_package	SN74LS08N.pbo
SN74LS13	SN74LS13_SN74LS13	SN74LS13_package	SN74LS13.pbo
SN74LS138	SN74LS138_SN74LS138	SN74LS138_package	SN74LS138.pbo
SN74LS138N	SN74LS138N_SN74LS138N	SN74LS138N_package	SN74LS138N.pbo
SN74LS139	SN74LS139_SN74LS139	SN74LS139_package	SN74LS139.pbo
SN74LS139N	SN74LS139N_SN74LS139N	SN74LS139N_package	SN74LS139N.pbo

Total Count: 92 Search Count: 92

Description of component: part name_package

Edit Component

Component Name: SN74LS08_package

Component Type: Parts

Gate Count: 1

Equivalence: 1=2, 4=5, 8=10, 12=13

Block File Name: [Icon]

Function Name: 2AND

Symbol File Assignment: Positive Symbol, Negative Symbol

Symbol Pin ID	Common Terminal? isCommonTerminal	Pin Number pinNumber	Pin Label pinLabel	IO io
1	NO	1	1A	-
2	NO	2	1B	-
3	NO	3	1Y	-
4	NO	4	2A	-
5	NO	5	2B	-
6	NO	6	2Y	-
7	NO	7	GND	GND
8	NO	8	3Y	-
9	NO	9	3A	-
10	NO	10	3B	-
11	NO	11	4Y	-
12	NO	12	4A	-
13	NO	13	4B	-
14	NO	14	VCC	VCC

Buttons: Insert Pin, Append Pin, Delete Pin, OK, Cancel



Rules for creating a power supply box (Component Type: Power Box)

Created only when the POWER, GROUND pin exists in the external terminal information.

Part name -> cdbName (ID-name)
partName

Symbol information -> Function name (symbol file name)

External pin number -> pinNumber

```

$PMA {
  NAME SN74LS08 : MODULE : TTL : 1 : 4 : 14 ;
  PCMAC_NO 5014:6014::;
  SYMBOL {
    AND2 : 1, 2, 3, 4 : A, B, Y : A = B ;
  }
  PIN {
    1 : 1A : 1 : A : ;
    2 : 1B : 1 : B : ;
    3 : 1Y : 1 : Y : ;
    4 : 2A : 2 : A : ;
    5 : 2B : 2 : B : ;
    6 : 2Y : 2 : Y : ;
    7 : GND : GROUND : GND : ;
    8 : 3Y : 3 : Y : ;
    9 : 3A : 3 : A : ;
    10 : 3B : 3 : B : ;
    11 : 4Y : 4 : Y : ;
    12 : 4A : 4 : A : ;
    13 : 4B : 4 : B : ;
    14 : +5V : POWER : +5V : ;
  }
  CURRENT 40.0
}
    
```

External pin name -> pinLabel

POWER, GROUND -> io property (VCC, GND)



LCDB Editor

d:/users/LCDB_lib/pmaster.prf

CDB Name	Component Name	Component Name	Use with Layout	Use for
	componentName	componentName	useWithLayout	useInPa
SN74LS08	SN74LS08_package	SN74LS08.pbox	--	--
SN74LS08M	SN74LS08M_package	SN74LS08M.pbox	--	--
SN74LS13	SN74LS13_package	SN74LS13.pbox	--	--
SN74LS13E	SN74LS13E_package	SN74LS13E.pbox	--	--
SN74LS138N	SN74LS138N_package	SN74LS138N.pbox	--	--
SN74LS139	SN74LS139_package	SN74LS139.pbox	--	--
SN74LS139N	SN74LS139N_package	SN74LS139N.pbox	--	--

Total Count: 92 Search Count: 92

Description of the component: part name.pbox

Edit Component

Component Name: SN74LS08.pbox

Component Type: Power Box

Gate Count: 1

Symbol File Assignment: Positive Symbol

Symbol Pin ID	Common Terminal? isCommonTerminal	Pin Number pinNumber	Pin Label pinLabel	IO io
1	NO	7	GND	GND
2	NO	14	VCC	VCC

Buttons: Insert Pin, Append Pin, Delete Pin, OK, Cancel



Conversion rules for symbol pins

Values that are converted into IO properties

IO properties	Pin Label
POWER ->	VCC
GROUND ->	GND
NOCONNECT ->	NC
Common signal ->	*Converted only when IO_CURRENT (Pin property information) is specified. If not, the pin Label is not specified.
	I (Input) -> INPUT
	O (Output) -> OUTPUT
	IO (Input/Output) -> BIDIRECT

Conversion of IO_CURRENT (Pin property information)

The maximum input/output current of each pin specified in the pin property information (IO_CURRENT) will be converted only when a property item is defined in the LCDB Parameter File (.prf). If not, the pin property information (IO_CURRENT) will not be converted.

HIGH level maximum input current ->	highInCurrent
HIGH level maximum output current ->	highOutCurrent
LOW level maximum input current ->	lowInCurrent
LOW level maximum output current ->	lowOutCurrent

```

$PMA {
NAME      0.10U : DISCRETE : : 1 : 1 : 2 ;
SYMBOL {
          0.1U : 1 : T1, T2 : T1=T2 ;
}
PIN {
  1 : T1 : 1 : T1 ;
  2 : T2 : 1 : T2 ;
}
CURRENT 0.02;
IO_CURRENT {
  CAPA : T1(IO) : 0.001 : 0.001 : 0.003 : 0.003;
  CAPA : T2(IO) : 0.001 : 0.001 : 0.003 : 0.003;
}
}
$PMA {
NAME      SN54LS77 : MODULE : TTL : 1 : 4 : 14 ;
SYMBOL {
          SN74LS77 : 1,2,3,4 : G,D,Q ;;
}
PIN {
  1 : 1D : 1 : D ;
  2 : 2D : 2 : D ;
  3 : EN3-4 : 3,4 : G ;
  4 : VCC : POWER : VCC ;
  5 : 3D : 3 : D ;
  6 : 4D : 4 : D ;
  7 : NC1 : NOCONNECT : NC ;
  8 : 4Q : 4 : Q ;
  9 : 3Q : 3 : Q ;
  10 : NC2 : NOCONNECT : NC ;
  11 : GND : GROUND : GND ;
  12 : EN1-2 : 1,2 : G ;
  13 : 2Q : 2 : Q ;
  14 : 1Q : 1 : Q ;
}
}
    
```

lowInCurrent	highOutCurrent	lowOutCurrent
0.001	0.003	0.003
0.001	0.003	0.003

Terminal?	Pin Number	Pin Label	IO
Terminal	pinNumber	pinLabel	io
	1	1D	-
	2	2D	-
	3	EN3_4	-
	4	VCC	VCC
	5	3D	-
	6	4D	-
	7	NC1	NC
	8	4Q	-
	9	3Q	-
	10	NC2	NC
	11	GND	GND

Converting user-defined properties

In a Common Device Database File (.pcp), user-defined items are specified in a user definition file (usdef), and their property values are specified in the Common Device Database File (.pcp).

pmaster.pcp

```

}
PIN {
  1 : T1 : 1 : T1 : ;
  2 : T2 : 1 : T2 : ;
}
USER_DEFINITION {
  1 : "RNA-330" ; /*PACKAGE_NAME*/
  2 : "R00002" ; /*PACKAGE_MODEL*/
  5 : "120" ; /*PACKAGE_PRICE*/
  20 : "10" ; /*POWER*/
}
    
```

usdef

```

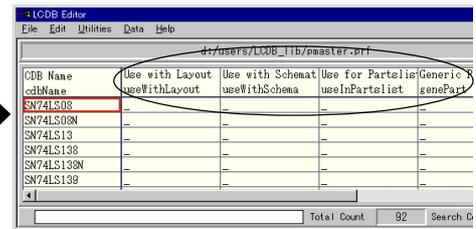
1 : PACKAGE_NAME : c :
2 : PACKAGE_MODEL : c :
3 : PACKAGE_TYPE : c :
4 : PACKAGE_MAKER : c :
5 : PACKAGE_PRICE : i : \
20 : POWER : f : mW
    
```

In a Component Database for Schematic Design (LCDB), user-defined items are specified in a LCDB Parameter File (.prf), and their property values are specified in a Database File (.dbf) and a Component Information File (.cmp).

sws.prf

```

cdbldName : partName
dataBaseFile : sws.dbf
componentFile : sws.cmp
device 2 {
  cdbName      text
  partName     text
  useWithLayout text
  useWithSchema text
  useInPartlist text
  genePart     text
  componentName text
  componentName text
  componentName text
  componentName text
  componentName text
  PACKAGE_NAME text
  PACKAGE_MODEL text
  PACKAGE_TYPE text
  PACKAGE_MAKER text
  PACKAGE_PRICE int
  POWER        int
}
cpin 2 {
  isCommonTerminal text
  pinNumber         text
    
```



When a Common Device Database File (.pcp) is converted to LCDB(Component Database for Schematic Design), all the properties in a specified user definition file (usdef) will be converted if a LCDB Parameter File (.prf) is not specified.

If properties that are not required to convert to a Component Database for Schematic Design (LCDB) exist, even if the properties are specified in the Common Device Database File (.pcp), you can convert only the necessary properties by creating its LCDB Parameter File (.prf).

Converting symbol names

In a Common Device Database File (.pcp), a section for specifying a symbol file on the schematic for the part does not exist. Therefore, since symbol file names are not specified when the Common Device Database File (.pcp) is converted into the Component Database for Schematic Design (LCDB), you need to specify the symbol file names one by one.

The [Symbol Name Auto Generate Mode] in the “Option” converts a symbol name in the SYMBOL section within the Common Device Database File (.pcp) into a gate component as a symbol file name used on the schematic.

```

$PMA {
  NAME          0.10U : DISCRETE : : 1 : 1 : 2 ;
  SYMBOL {
    0.1U : 1 : T1, T2 : T1=T2 ;
  }
  PIN {
    1 : T1      : 1      : T1      ;
    2 : T2      : 1      : T2      ;
  }
  CURRENT 0.02;
  IO_CURRENT {
    CAPA   : T1(IO) : 0.001 : 0.001 : 0.003 : 0.003 ;
    CAPA   : T2(IO) : 0.001 : 0.001 : 0.003 : 0.003 ;
  }
}

```



When “symbol” is set:



When “OFF” is set:

Positive Symbol	Negative Symbol
0.1U.smb	-

Positive Symbol	Negative Symbol

Converting component numbers (PCMAC_NO)

When you convert a component number (PCMAC_NO) that is being enabled in the Common Device Database File (.pcp), specify in advance the following user-defined properties when you create a Component Database for Schematic Design (LCDB).

- PCMAC1
- PCMAC2
- PCMAC3
- PCMAC4
- PCMAC5

LCDB Parameter File (.prf)

```

:      :
componentName      text
componentName      text
componentName      text
componentName      text
componentName      text
PACKAGE_NAME      text
PACKAGE_MODEL      text
:      :
PCMAC1             text
PCMAC2             text
PCMAC3             text
PCMAC4             text
PCMAC5             text
:      :
    
```

PCMAC_NO : (1) : (2) : (3) : (4) : (5) ;

```

$PMA {
NAME      0.10U : DISCRETE : : 1 : 1 : 2 ;
PCMAC_NO  5102: 6102 : : ;
SYMBOL {
0.1U : 1 : T1, T2 : T1=T2 ;
}
PIN {
1 : T1 : 1 : T1 ;
}
    
```

CDB Name	PWS PCMACRO NO 1	PWS PCMACRO NO 2	PWS PCMACRO NO 3	PWS PCMACRO NO 4	PWS
cdBName	PCMAC1	PCMAC2	PCMAC3	PCMAC4	PCM
0.1U	5102	6102	-	-	-
0.33MF	1008	-	-	-	-
1.8K	5101	6101	-	-	-
220	5101	6101	-	-	-
2764	5216	6816	-	-	-
2MHZ	5302	6802	-	-	-

stored into file (d:/users/LCDB_lib/pmaster.prf). Total Count 94 Search Count 94

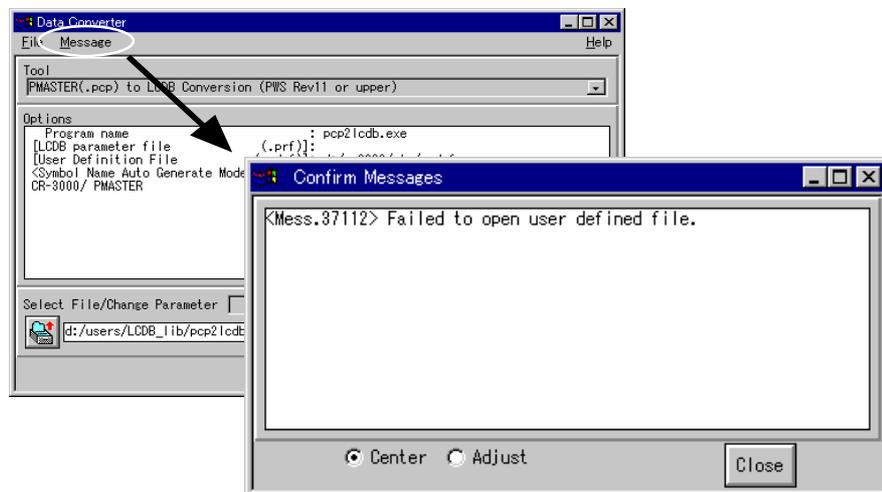
Displaying Warning/Error Messages

Warning or error messages displayed at conversion are created in the directory where either a file or a directory specified by a “Option” as a file converted exists. Each description is as follows:

Error message: pcp2lcdb.err

Warning message: pcp2lcdb.wrn

The messages can be confirmed by Data Converter.
Select [Message]-[Error] or [Message]-[Warning] from the menu bar.



Starting from Shell (Only for UNIX version)

The PMASTER (.pcp) to LCDB Conversion Program can be started from the shell.

```
% pcp2lcdb.sh [option] filename
```

```
options : [-r User-Definition (usdef)]           ... User definition file
          [-r Prf-File-Name (.prf)]             ... LCDB Parameter File
          [-V] --- output version number        ... (Version check)
mode : [-m symbol] --- output symbol name      ... Conversion mode of a symbol name
filename: Target-Pmaster (.pcp)                ... File converted
```

```
Example: %pcp2lcdb.sh /users/s2s/sws.pcp
```

```
Example: %pcp2lcdb.sh -r /users/s2s/lcdb.prf -m symbol /users/s2s/sws.pcp
```

● **Precautions**

 **Reference** Also see the online help of Data Converter.

<Conversion List for PMASTER -> LCDB>

pmaster	LCDB	Remarks
Device Name	cdbName PartName (Part name)	
Component Number information 1~ 5	PCMAC1 ~ 5	Converted only when the item is provided in the device property.
Symbol Name	function (Function name)	It will not be converted to package symbols for a Power Box and for parts with different gates.
Terminal Equivalency Definition	Equivalence	
Pin Number	pinNumber	
Pin Name Terminal Name	pinLabel pinLabel	Component Type: The external pin name will be pinLabel only for the parts component.
VCC,GND,NOCONNECT	io	
Terminal Name(I/O Attribute)	io	
Maximum input current, HIGH level Maximum input current, LOW level Maximum output current, HIGH level Maximum output current, LOW level	highInCurrent lowInCurrent highOutCurrent lowOutCurrent	Converted only when the item is provided in the LCDB component property.
User-Defined Item	User-defined device property	Converted only when the same label described in usdef is provided in the device property.

<Unavailable data conversion>

pmaster	LCDB
Part level Logic Maximum Supply Current of parts	blockName

<Restrictions> * Since LCDB cannot handle 'en'-size kana, they are converted to 'em'-size kana.
 * If the number of components generated from one device is greater than that of "componentName" provided in the LCDB Parameter File (.prf), the processing for that device is skipped. ("<Mess. 37118> Cannot write into the dbf file due to the shortage of columns for (componentName) in the prf file.")

```

$PMA {
  NAME SN74LS31 : MODULE :TTL : 3 : 6 : 16 ;
  SYMBOL {
    INV : 1,6 : A, Y : ;
    BUF : 2,5 : A, Y : ;
    2NAND : 3,4 : A, B, Y : A = B ;
  }

```

Creates Number of available gate types + Two components
 Gate x the number of types (part name_symbol name)
 Component package (part name_package)
 Power supply box (part name_pbox)

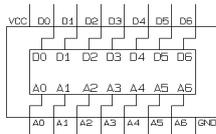
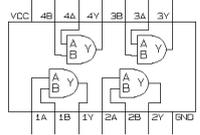
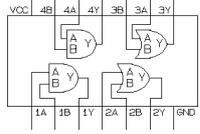
↓

SN74LS31_INV, CSN74LS31_BUF, CSN74LS31_2NAND, CSN74LS31_package, CSN74LS31.pbox

In case of the above example, you must specify at least five componentName in the LCDB Parameter File (.prf)

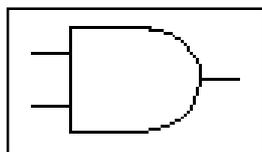
<About component information>

The component information (componentName) to be created changes as follows depending on parts registered in PMASTER.

Parts information (PMaster)	LCDB component	Component name
1 package, 1 gate 	<ul style="list-style-type: none"> • Gate 	part name_symbol name part name_package
1 package, 1 gate 	<ul style="list-style-type: none"> • Gate • Part 	part name_symbol name part name_package part name_pbox
1 package, multiple gates 	<ul style="list-style-type: none"> • Gate • Part • Power Box 	part name_symbol name part name_package part name_pbox
1 package, multiple gates 	<ul style="list-style-type: none"> • Gate (for the number of types) • Part • Power Box 	part name_symbol name part name_package part name_pbox

When the parts with POWER, GRAND pin are converted, the POWER, GRAND pin is added at the back of the gate pin.

Example: AND2



Symbol pin ID	isCommonT	pinNumber	pinLabel
1	NO	1,4,9,12	A
2	NO	2,5,10,13	B
3	NO	3,6,8,11	Y
4	YES	7,7,7,7	GND ←
5	YES	14,14,14,14	VCC ←

Pin ID Consistency Check

In SWS, pin numbers are assigned in the order in which the pin was entered into the symbol file. (In SWS, the pin number is not specifically taken into account except when AWS is used.) The pin number is converted as a symbol pin ID when it is converted to System Designer.

When you input components with LCDB in System Designer, the pin property specified in LCDB is assigned by using a pin ID for the symbol as a keyword. Therefore, if a symbol pin ID and a pin ID specified in LCDB is not consistent, a schematic will be created with a wrong pin information.



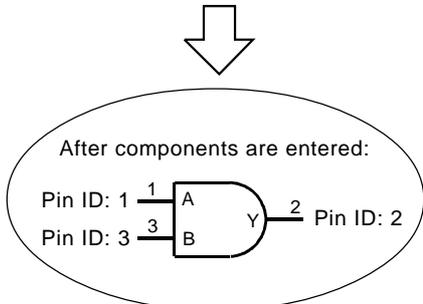
PMASTER

```

$PMA {
NAME SN74LS08 : MODULE : TTL : 1 : 4 : 14 ;
PCMAC_NO 5014:6014:;;;
SYMBOL {
    AND2 : 1, 2, 3, 4 : A, B, Y : A = B ;
}
PIN {
1 : 1A : 1 : A ;
2 : 1B : 1 : B ;
3 : 1Y : 1 : Y ;
4 : 2A : 2 : A ;
5 : 2B : 2 : B ;
6 : 2Y : 2 : Y ;
7 : GND : GROUND : GND ;
8 : 3Y : 3 : Y ;
9 : 3A : 3 : A ;
10 : 3B : 3 : B ;
11 : 4Y : 4 : Y ;
12 : 4A : 4 : A ;
13 : 4B : 4 : B ;
14 : +5V : POWER : +5V ;
}
}
    
```

Component Database for Schematic Design (LCDB)

	Pin name	Pin number
pin ID1	A	1,4,9,12
pin ID2	B	2,5,10,13
pin ID3	Y	3,6,8,11
pin ID3	GND	7
pin ID3	VCC	14



The program that checks the inconsistency between a symbol pin ID and a pin ID of LCDB (Component Database for Schematic Design), and rewrites the pin ID of the symbol file to be consistent with the pin ID of LCDB is the "PMaster and System Designer Data Pin ID Check and Adjust".

This consistency check comes in useful when:

1. The Common Device Database File (pmaster.pcp) for PWS/WS is converted into LCDB (Component Database for Schematic Design), and a symbol sheet of SWS is converted into a symbol sheet of CR-5000/System Designer (xx.sht).

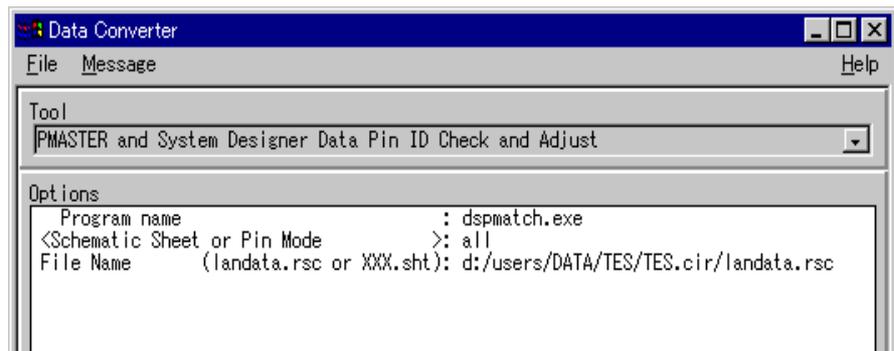
This tool checks consistency between the pin assignment of a symbol used in the symbol sheet and the pin assignment of LCDB (Component Database for Schematic Design), and rewrites the pin ID of the symbol sheet. (This is effective for management of libraries.)

2. A schematic sheet for SWS is converted into the Circuit Directory (xx.cir) and an operation using a Common Device Database File (pmaster.pcp) is performed at SWS.

This tool checks consistency between the pin assignment of the symbol sheet and the pin assignment of LCDB (Component Database for Schematic Design), and rewrites the pin ID of the symbol, and replaces the symbol on the schematic sheet with the correct symbols.

Start

[PMASTER and System Designer Data Pin ID Check and Adjust] from "Data Converter".



Options

- 1) [Program name]: (dspmatch.exe) Cannot be modified
- 2) [Schematic Sheet or Pin mode]: Specifies that the result of the check is applied to a schematic sheet when the schematic sheet is checked.
 - all -> After the pin IDs in the symbol sheet are rewritten, the symbols in the schematic sheet are replaced with the correct symbols to function as regular symbols.
 - pin -> Rewrites only the pin IDs in the symbol sheet.
- 3) [File name]: Specifies either the Data resource file (landata.rsc) or the schematic sheet.

landata.rsc	->	Checks consistency between the symbols and LCDB (Component Database for Schematic Design). (Relative path is NG.)
xxx.sht	->	Checks consistency between the symbols and LCDB (Component Database for Schematic Design). (Relative path is OK.)

- PMASTER and System Designer Data Pin ID Check and Adjust uses the pin Label of LCDB (Component Database for Schematic Design) and the pin Label in the symbol name, so be sure to input a pin Label in the symbol.
- Schematic sheets are checked based on the "part name" assigned to the component, so be sure to input a part name.



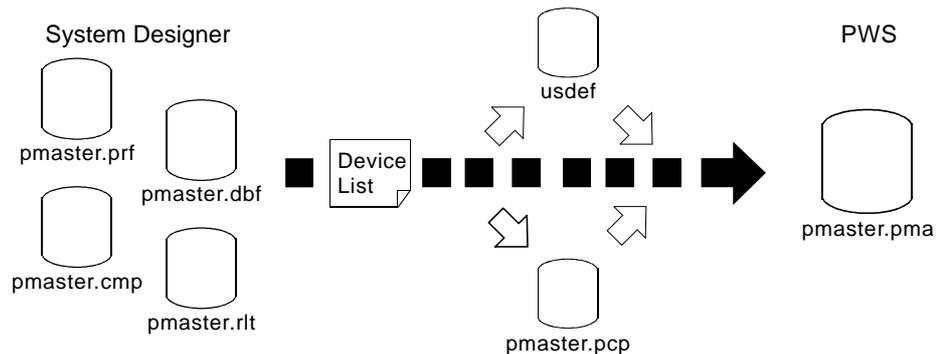
If an error occurs, the message will be displayed in \$HOME/cr5000/ds/log/dspmatch.err.
If a warning occurs, the message will be displayed in \$HOME/cr5000/ds/log/dspmatch.wrn.

2. LCDB to PMASTER (PMA)

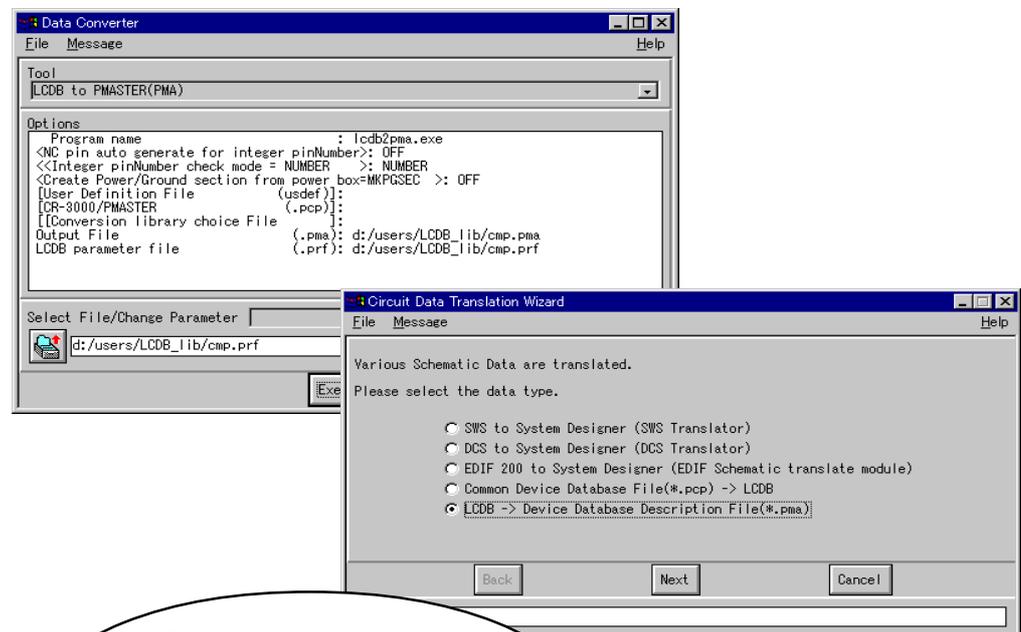
● LCDB to PMASTER (PMA)

It converts LCDB (Component Database for Schematic Design) created by CR-5000/System Designer into a Component Property Definition File (.pma).

! CAUTION To use the converted data as a Common Device Database File (.pcp), you need to run [pmreg].



Select [LCDB to PMASTER (PMA)] from the option list of “Tool” in [Data Converter].



In [Circuit Data Translation Wizard], select [LCDB -> Device Database Description File (*.pma)].



For information on a common device database file, see "PWS Design Preparation Files - Device Database Description File Reference File (.pma)".

Specify the following startup options. (Some of the options are optional.)

Options

Program name: Cannot be modified

UNIX:(lcdb2pma.sh) Windows:(lcdb2pma.exe)

[NC pin auto generate for integer pinNumber]Default: OFF

If this value is set to "NCGEN", PMA automatically generates omitted numbers as NOCONNECT (NC) pins when the pin numbers defined by only figures in LCDB are not serial.

[Integer pinNumber check mode] Default: OFF

If this value is set to "NUMBER", pin numbers are checked if it is consisted only of numeric values. If characters other than numeric are detected, a warning message will be displayed.

[Create Power / Ground section from power box] Default: OFF

If this value is set to "MKPGSEC", the pin information for a Power Box is considered as POWER/GROUND information for PMA. Pins with the power/ground properties of parts without a Power Box should be general terminals or common terminals instead of assigning "POWER/GROUND" property to the pins.

If this value is not set to "MKPGSEC", all the pins that have the properties of a power/ground should have the "POWER/GROUND" property assigned.

[User definition file (usdef)]: Optional

If the property you want to use as a user-defined item in PMASTER is defined as a user-defined property in LCDB, specify the file in which its item name is defined. If the file is not specified, it will not be converted even if the user-defined item is specified in the LCDB parameter file (.prf).

\$CRLOCAL/cr3000/etc/usdef

```
1:partNumber:c:
2:MAKER:c:
3:PRICE:i:\
```

[CR-3000 / PMASTER (.pcp)]: Optional

This option is specified when only parts not registered in the existing PMASTER are converted. If the file is not specified, all the parts in LCDB will be converted.

[Conversion library choice File]: Optional

This option creates a list of part names and specifies their file name when parts to be converted are restricted. If the file is not specified, all the parts in LCDB will be converted.

```
SN74LS00
SN74LS04
SN74LS08
:
```

Write only one cdbName in each line.

[Output File]: Required

This option specifies the file name to be displayed. If you add .pma to its extension, it is useful when the file is converted into a Common Device Database File (.pcp). If the existing file name is specified, the file will be overridden.

[LCDB parameter file (.prf)]: Required

This option specifies LCDB to be converted. Specify the file with .prf extension.

After specifying the Options, press the <<Execute>> button.

● **About Conversion Rules**

Component Property Definition File (.pma) Conversion is performed as follows depending on the settings of each parts in LCDB.

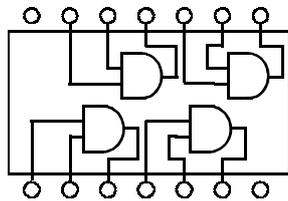


LCDB allows you to specify multiple components for each parts, but the Common Device Database File (.pcp) allows you to specify only one component for one part (gate or device).

1. For parts that have gate components

If parts either have gate components only, or both the package components and the gate components, the gate components other than the Power Box is converted.

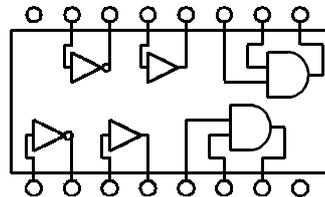
If the gates are of a single type:



SN74LS08_package (package)	SN74LS08_2AND (gate)	SN74LS08_pbox (power box)
-------------------------------	-------------------------	------------------------------

```
$PMA{
  NAME SN74LS08 : MODULE :TTL:1:4:14;
  PCMAC_NO 5014 : 6014 : : : ;
  SYMBOL{
    2AND :1,2,3,4: A,B,Y :A=B ;
```

If the gates are of multiple types:



SN74LS31_package (package)	SN74LS31_INV (gate)	SN74LS08_BUF (gate)
SN74LS31_2AND (gate)	SN74LS31_pbox (power box)	

```
$PMA{
  NAME SN74LS31 : MODULE :TTL:3:6:16;
  PCMAC_NO 5016 : 6016 : : : ;
  SYMBOL{
    INV : 1,6 : A , Y : ;
    BUF : 2,5 : A , Y : ;
    2NAND : 3,4 : A,B,Y :A=B;
```

2. For parts that have package components only

If parts have package components only (except the power box), only the package components are converted.

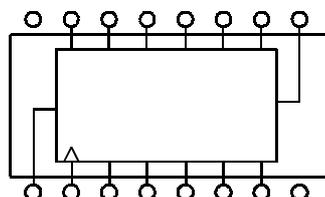
When only package components exist:



0.3U_package (package)		
---------------------------	--	--

```
$PMA{
  NAME 0.3U : DISCRETE : :1:1:2;
  PCMAC_NO 5102 : 6102 : : : ;
  SYMBOL{
    0.3U : 1 : Q1,Q2 : Q1=Q2 ;
```

When package and power box exist:



SN74LS165_package (package)	SN74LS165_pbox (power box)	
--------------------------------	-------------------------------	--

```
$PMA{
  NAME SN74LS165:MODULE:ELSE:1:1:16;
  SYMBOL{
    SN74LS165 : 1 :SL,CK,E,F,G,H,
    %QH%,QH,SI,A,B,C,D,CI : ;
```



Rules for creating PMA (Component Type: When Gate exist)

IDname	partName	partNumber	ComponentName	ComponentName	ComponentName
SN74LS08	SN74LS08	Z121A1121	AND2*4_package	AND2*4_AND2	AND2*4_pbox
4.7K	4.7K	Z409S7383	RESI_package		
:	:	:	:	:	:

AND2*4_package

AND2*4_AND2

PCMAC1	PCMAC2	MAKER
5014	6014	ZUKEN
102	202	ZUKEN
:	:	:

Component type: Part
The number of gates 1
Equivalence
Function name

Component type: Gate
The number of gates 4
Equivalence 1=2
Function name: AND 2

Symbol pin ID	pinNumber	pinLabel
1	1	1A
2	2	1B
:	:	:
13	13	4B
14	14	VCC

Symbol pin ID	pinNumber	pinLabel
1	1,4,9,12	A
2	2,5,10,13	B
3	3,6,8,11	Y
4	7	GND
5	14	VCC



Refer to the user definition file (usdef) and perform a conversion

\$CRLOCAL/cr3000/etc/usdef

```
1:MAKER:c:
2:partNumber:c:
```



Device Name <- partName

Component Level = conversion disabled
Logic = conversion disabled

Component Number
<- PCMAC1-5

Symbol Name
<- Function name

Pin number
<- pinNumber

External pin name

- When package components exist
<- pinLabel of package components
- When only gate components exist
<- PIN + Serial number
(Auto-generation)

Maximum Supply Current
= conversion disabled

```
$PMA {
  NAME SN74LS08 : MODULE : ELSE : 1 : 4 : 14 ;
  PCMAC_NO 5014:6014::;
  SYMBOL {
    AND2 : 1, 2, 3, 4 : A, B, Y : A = B ;
  }
  PIN {
    1 : 1A : 1 : A ;
    2 : 1B : 1 : B ;
    3 : 1Y : 1 : Y ;
    4 : 2A : 2 : A ;
    5 : 2B : 2 : B ;
    6 : 2Y : 2 : Y ;
    7 : GND : GROUND : GND ;
    8 : 3Y : 3 : Y ;
    9 : 3A : 3 : A ;
    10 : 3B : 3 : B ;
    11 : 4Y : 4 : Y ;
    12 : 4A : 4 : A ;
    13 : 4B : 4 : B ;
    14 : +5V : POWER : +5V ;
  }
  CURRENT 40.0 POWER,GROUND <- io property (only VCC,GND)
  USER_DEFINITION {
    1 : "ZUKEN" ; /* MAKER */
    2 : "Z121A1121" ; /* partNumber */
  }
}
```



Rules for creating PMA (Component types: When only parts exist)

IDname	partName	partNumber	ComponentName	ComponentName	ComponentName
SN74LS08	SN74LS08	Z121A1121	AND2*4_package	AND2*4_AND2	AND2*4_pbox
4.7K	4.7K	Z409S7383	RESI_package	-	-
:	:	:	:	:	:

RESI_package

Component type: parts
 The number of gates 1
 Equivalence 1=2
 Function name: RES1

Symbol pin ID	pinNumber	pinLabel
1	1	P1
2	2	P2

PCMAC1	PCMAC2	MAKER
5014	6014	ZUKEN
102	202	ZUKEN
:	:	:

↓

Refer to the user definition file (usdef) and perform a conversion
 \$CRLOCAL/cr3000/etc/usdef

1:MAKER:c:
 2:partNumber:c:

↓

Device Name <- partName Logic = conversion disabled
 Component Level = conversion disabled

```

$PMA {
  NAME 4.7K : MODULE : ELSE : 1 : 1 : 2 ;
  PCMAC_NO 5014:6014::;
  SYMBOL {
    RESI : 1, 2 : P1, P2 : P1 = P2 ;
  }
  PIN {
    1 : P1 : 1 : P1 ;
    2 : P2 : 1 : P2 ;
  }
  CURRENT 40.0
  USER_DEFINITION {
    1 : "ZUKEN" ; /* MAKER */
    2 : "Z409S7383" ; /* partNumber */
  }
}
        
```

Component Number <- PCMAC1-5

Symbol Name <- Function name

Pin Number <- pinNumber

Pin Name <- pinLabel

Maximum Supply Current = conversion disabled

Pin name -> pinLabel



Items that cannot be converted:

User definition properties assigned to pins
 Maximum current consumption (CURRENT)

About conversion of user definition properties

In LCDB (Component Database for Schematic Design), user-defined items are specified in a LCDB Parameter File (.prf), and their property values are specified in a Database File (.dbf) and a Component File (.cmp). There is no limitations on the number of items.

sws.prf

```

cdbldName : partName
dataBaseFile : sws.dbf
componentFile : sws.cmp
device 2 {
    cdbName          text
    partName         text
    useWithLayout    text
    useWithSchema    text
    useInPartlist    text
    genePart        text
    componentName   text
    componentName   text
    componentName   text
    componentName   text
    componentName   text
    PACKAGE_NAME    text
    PACKAGE_MODEL   text
    PACKAGE_TYPE    text
    PACKAGE_MAKER   text
    PACKAGE_PRICE   int
    POWER           int
}
cpin 2 {
    isCommonTerminal text
    pinNumber        text

```

CDB Name	Use with Layout	Use with Schematic	Use for Partlists
cdbName	useWithLayout	useWithSchema	useInPartlist
SN74LS08	-	-	-
SN74LS08N	-	-	-
SN74LS13	-	-	-
SN74LS13B	-	-	-
SN74LS138N	-	-	-
SN74LS139	-	-	-

Since user-defined items are specified in a user definition file (usdef) in a Common Device Database File (.pcp), the number of items is limited to 20.

The number of user definition property that can be registered in a Common Device Database File (.pcp) is limited to 20 items. Therefore, when you make a conversion, specify only the necessary user definition property in the user definition file (usdef).

usdef

```

1 : PACKAGE_NAME : c :
2 : PACKAGE_MODEL : c :
3 : PACKAGE_TYPE : c :
4 : PACKAGE_MAKER : c :
5 : PACKAGE_PRICE : i :
20: POWER : f :mW

```

RESI.pma

```

}
PIN {
  1 : T1 : 1 : T1 ;
  2 : T2 : 1 : T2 ;
}
USER_DEFINITION {
  1:"RNA-330" /*PACKAGE_NAME*/
  2:"R00002" /*PACKAGE_MODEL*/
  5:"120" /*PACKAGE_PRICE*/
  20:"10" /*POWER*/
}
}

```

About conversion of component number (PCMAC_NO)

When you convert a component number (PCMAC_NO) that can be specified in a Common Device Database File (.pcp), specify the following user definition property when you create LCDB (Component Database for Schematic Design).

PCMAC1
PCMAC2
PCMAC3
PCMAC4
PCMAC5

LCDB Parameter File (.prf)

```

:      :
componentName  text
componentName  text
componentName  text
componentName  text
componentName  text
PACKAGE_NAME  text
PACKAGE_MODEL  text
:      :
PCMAC1        text
PCMAC2        text
PCMAC3        text
PCMAC4        text
PCMAC5        text
:      :
    
```

PCMAC_NO : (1) : (2) : (3) : (4) : (5);

LCDB Editor

d:/users/LCDB_lib/pmaster.prf

CDB Name	PWS PCMACRO NO 1	PWS PCMACRO NO 2	PWS PCMACRO NO 3	PWS PCMACRO NO 4	PWS
cdName	PCMAC1	PCMAC2	PCMAC3	PCMAC4	PCM
0.1U	5102	6102	-	-	-
0.33MF	1008	-	-	-	-
1.8K	5101	6101	-	-	-
220	5101	6101	-	-	-
2764	5216	6816	-	-	-
2MHZ	5302	6802	-	-	-

stored into file (d:/users/LCDB_lib/pmaster.prf). Total Count 94 Search Count 94

```

$PMA {
  NAME      0.10U : ELSE : : 1 : 1 : 2 ;
  PCMAC_NO  5102: 6102 : : : ;
  SYMBOL {
    0.1U : 1 : T1, T2 : T1=T2 ;
  }
  PIN {
    1 : T1 : 1 : T1 ;
  }
}
    
```

Outputting Warning/Error Messages

Warning or error messages displayed at conversion are created in the directory where either a file or a directory specified by a "Option" as a file converted exists. Each description is as follows:

Error message : lcdb2pma.err
Warning message : lcdb2pma.wrn

The messages can be confirmed by the Data Converter.
Select [Message]-[Error] or [Message]-[Warning] from the menu bar.

Starting from Shell (Only for UNIX version)

Pmaster(.pcp) to LCDB Conversion Program can be started from the shell.

% *lcdb2pma.sh [option] filename*

options :	[-m NCGEN]	...
	[-m NUMBER]	...
	[-r User-Definition (usdef)]	... User definition file
	[-r Pmaster (.pcp)]	... Common Device Database File (.pcp)
	[-p P-File]	... Conversion device specification file
	[-o Output-File-Name]	... Output file
	[-V] --- output version number	... (Version check)
filename:	Target-Prf-File-Name (.prf) or	... File converted
	Target-Cir-Name (.cir)	

Example: %lcdb2pma.sh -o /users/s2s/sws.pma /users/sws/lcdb.prf

Example: %lcdb2pma.sh -p /users/s2s/newparts -o /users/s2s/sws.pma /users/s2s/lcdb.prf

About conversion from Component Property Definition File Conversion (.pma) to PMASTER

A conversion from Component Property Definition File Conversion (.pma) to PMASTER is performed on a machine on which PWS operates. Use the "Device Database Management Tool Operation Manual" or the <pmreg> program.

To newly create and add PMASTER:

<XX> pmreg -c lcdb ↓
New create mode PMASTER name to be created

<XX> pmreg -m lcdb lcdb ↓
Exclusive PMASTER assignment Converted PMA file name

To override and add to the existing PMASTER:

<XX> pmreg -r -m lcdb lcdb ↓
Update mode Exclusive PMASTER assignment Converted PMA file name

● **Precautions**

<Conversion List for LCDB -> PMASTER>

LCDB	PMA	Restrictions on Characters
Device property partName (Part name) PCMAC 1~5 User definition property	Device Name ComponentNumberinformation1~5 User-defined Item	20 alphanumerics or less 1~12288 20 items or less
Component information function (Function name) The total number of gates in a gate component Equivalence	Symbol Name Number of gates Terminal Equivalency Definition	20 alphanumerics or less <div style="border: 1px solid black; border-radius: 50%; padding: 5px; display: inline-block;">all 'en'-size alphanumerics except for user-defined items</div>
Pin property Pin number of part (or gate) component Pin Label of part (or gate) component Pin Label of gate component I/O property highInCurrent lowInCurrent highOutCurrent lowOutCurrent	External pin number External pin name Pin name VCC, GND, NC, Input/output pin properties HIGH level maximum input current LOW level maximum input current HIGH level maximum output current LOW level maximum output current	Integer String including alphabets of 20 or less Alphanumerics of 20 or less Real number Real number Real number Real number

<Unavailable data conversion>

LCDB
cdbName componentName symbolName blockName User definition pin property

PMA	
Component Level	All "MODULE"
Logic	All "ELSE"
Maximum Supply Current	The whole section is omitted of parts

<Restrictions>

- ◆ Even if it is written in lowercase letters, PWS interprets the lowercase letters as uppercase letters. If no measures are taken, it may cause a problem at Back annotation. Therefore, we recommend you to modify LCDB. (See the warning message displayed at conversion.)
- ◆ Since LCDB to PMA Conversion Program creates PMA by converting the strings as-is. Therefore, it may cause an error when executing pmreg even if an error does not occur at conversion or when creating the board. If a warning message is displayed at the LCDB to PMA Conversion, you need to modify LCDB according to the message. (For example, redundancy of external pin names may cause an error when executing pmreg.)
- ◆ If parts without pins are registered in LCDB, convert the parts into PMA as well. (The parts will also be registered in PMASTER, but there is no need to refer to those parts when the board is designed.)