Rev. 1 - 11/2020

Application Note

1 Introduction

This document describes the steps required to configure pre-boot loader (PBL) on NXP QorIQ platform using the PBL tool included in QorIQ Configuration and Validation Suite (QCVS).

This document explains:

- · Purpose of the QCVS PBL tool
- · How to configure PBL using the PBL tool
- · PBL tool limitations

2 Preliminary background

The QCVS PBL tool provides you a graphical user interface (GUI) for editing a PBL binary in a decoded form. The PBL binary can be created from scratch, imported from an existing reset configuration word (RCW) memory dump (such memory dumps can be obtained using U-Boot), or created based on data read from target.

The RCW data contains reset configuration information that PBL loads from a memory device during power-on or hardware reset. All data read from the RCW source is written to the RCW status registers by PBL. If RCW selects pre-boot initialization (PBI), then the PBI commands are processed and routed to CCSR, DDR, and other memory spaces.

The PBL tool operates in the context of documented PBL configuration constraints and errata and prevents the user from violating them. The output of the PBL tool is a PBL binary that can be used to pre-program the platform.

3 Creating a QorIQ configuration project

Perform the following steps to create a QorIQ configuration project with the PBL tool:

- 1. Open the QCVS Eclipse integrated development environment (IDE).
- 2. Choose File > New > QorlQ Configuration Project from the IDE menu bar. The New QorlQ Configuration Project wizard starts, displaying the Create a QorlQ Configuration Project page.
- Specify the project name in the Project name text box, and click Next. The Devices page appears.
- 4. Choose a device and a device version, and click Next. The Toolset selection page appears.
- 5. Select the PBL Preboot Loader RCW configuration checkbox, and click Next. The PBL configuration page appears, where you can choose an initial PBL configuration for your project using one of the following three options, specify other required settings, and complete project creation:
 - · Create default configuration
 - Import configuration from an existing PBL file
 - Read configuration from target's Reset Configuration Word Status Registers



Contents

1	Introduction	1
2	Preliminary background	1
3	Creating a QorIQ configuration	
	project	1
4	Basic PBL operations	4
5	Advanced PBL operations	20
6	PBL tool limitations	21
Α	Revision history	21

3.1 Create default configuration

The default PBL configuration includes basic settings for RCW and no PBI commands. Use the **Create default configuration** option when neither you need to customize an existing RCW dump (see Import configuration from an existing PBL file) nor you want to start from the RCW read from target.

The figure below shows the PBL configuration page with the Create default configuration option selected.

PBL configuration					
Choose PBL configuration	on				
					^
Basic Configuration					
Create default co					
	tion from an existing PBL f	file			
Read from target					
Create default configura	ation for PBL.				
					~ -
?	< <u>B</u> ack	Next >	<u>F</u> inish	Cancel	

3.2 Import configuration from an existing PBL file

The **Import configuration from an existing PBL file** option allows you to import PBL from other projects/resources, such as SDK. This option is useful when you need to quickly investigate and/or customize an existing PBL.

The figure below shows the PBL configuration page with the Import configuration from an existing PBL file option selected.

🥬 New QorlQ Configuratio	on Project		_		×
PBL configuration					
Choose PBL configuration					
Basic Configuration					^
Create default confi	guration				
	n from an existing PBL f	ile			
O Read from target	_				
Import Configuration					- 11
Input file: c:\Products\CV	/4NET_v2020.06\ws\PBL	.bin		Browse.	
File format: Binary				,	~
?	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Cancel	
 2. Importing a configuration 	from an existing PBL fi	e			

3.3 Read configuration from target's Reset Configuration Word Status Registers

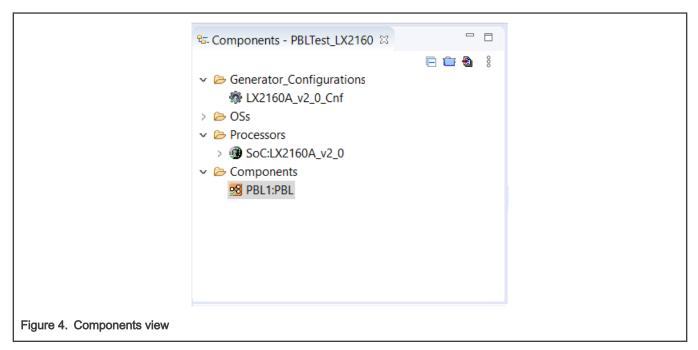
The existing RCW configuration can be used as the starting point for the PBL configuration. If you are new to the QCVS PBL tool, then you should use the **Read from target** option to create a QorlQ configuration project with the PBL tool. Target RCWs represent a good starting point to customize a PBL for a custom or reference design board.

The figure below shows the PBL configuration page with the Read from target option selected.

PBL configuration						
Choose PBL configuration						
Basic Configuration						^
Create default configura	tion					
O Import configuration fro	m an existing PBL	file				
Read from target						
Connection setting						
Probe type: CWTAP	✓ target_ip					
Read from target				Show De	etails	
Clock setting						
System clock: 100.0 MHz	∨ DDR clock:	100.0 MHz ~~				
Reads RCWSR registers from ta	arget.				~]
						-
٢					<u> </u>	
?	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	(Cancel	

4 Basic PBL operations

When you create a QorlQ configuration project with the PBL tool, a PBL component is created under the **Components** folder in the **Components** view, as shown in the figure below.



To view or edit the properties of the PBL component, select the PBL component in the **Components** view. The component properties are displayed on the **Properties** page of the **Component Inspector** view, as shown in the figure below.

NOTE

If the **Component Inspector** view is not open already, then open it by right-clicking a component in the **Components** view and choosing **Inspector** from the shortcut menu.

Properties	Import Validation			\checkmark	\$
Name		Value	Details		1
	MEM2_PLL_RAT [23 - 18]	0b011101 - 29:1 (Async Mo			
×	Cluster Groups PLL [143 - 24]				
×	Cluster Group A PLL [47 - 24]				
	DDR2_PLL_CLOCK	2.900 GT/s			
	CGA_PLL1_CFG [25 - 24]	0b00 - Default value			
	CGA_PLL1_RAT [31 - 26]	0b001111 - 15:1 (Async Mo			
	CGA_PLL1_CLOCK	1.500 GHz			
	CGA_PLL2_CFG [33 - 32]	0b00 - Default value			
	CGA_PLL2_RAT [39 - 34]	0b010100 - 20:1 (Async Mo			
	CGA_PLL2_CLOCK	2.000 GHz			
>	Cluster Group B PLL [127 - 48]				
	SYS_PLL_SPD [128 - 128]	0b0 - Default value			
	MEM_PLL_SPD [129 - 129]	0b0 - Default value			
	MEM2_PLL_SPD [130 - 130]	0b0 - Default value			
	CGA_PLL1_SPD [132 - 132]	0b0 - Default value			
	CGA_PLL2_SPD [133 - 133]	0b0 - Default value			
	CGB_PLL1_SPD [135 - 135]	0b0 - Default value			
	CGB_PLL2_SPD [136 - 136]	0b0 - Default value			
×	Cluster PLL Selection [175 - 144]				
	C1_PLL_SEL [147 - 144]	0b0000 - CGA_PLL1 / 1			
	C1_PLL_CLOCK	1.500 GHz			1
	C2_PLL_SEL [151 - 148]	0b0000 - CGn_PLL1 / 1			
	C2_PLL_CLOCK	1.500 GHz			
	C3_PLL_SEL [155 - 152]	0b0000 - CGn_PLL1 / 1			
	C3_PLL_CLOCK	2.000 GHz			
	C4_PLL_SEL [159 - 156]	0b0000 - CGn_PLL1 / 1			
	C4_PLL_CLOCK	2.000 GHz			
	C5_PLL_SEL [163 - 160]	0b0000 - CGn_PLL1 / 1			
	C5 PLL CLOCK	2.000 GHz			1

As you can see in the figure above, the properties of the PBL component are grouped under various categories and subcategories. The properties are sorted by the RCW position.

You can also notice that some of the component properties are grayed out. These are read-only properties; they cannot be changed. These properties are computed based on RCW fields.

Another useful feature of this presentation is that the most recently modified properties are displayed with yellow background. This way you can easily determine which other properties depend on the last modified properties.

Following are some basic PBL operations you can perform in the Component Inspector view:

- Change RCW bit field values
- Specify custom values
- · Display and set reserved fields
- Add PBI commands to a PBL image
- Import a PBL configuration from a file

- Import a PBL configuration from target
- Generate a PBL image
- Automatic PBL validation and errata support
- PBL validation tool
- Synchronize PBL with other IP blocks
- View RCW status registers

4.1 Change RCW bit field values

For most of the RCW fields, you can change the value by typing in a new value or choosing another value from a menu. For other fields (for example, SerDes protocol options), a more advanced graphical user interface (GUI) is displayed to change the value.

The figure below shows an example of changing a field value by choosing another value from a menu.

Properties	Import Validation			🖌 🗖 🔚 🔌
Name		Value	Details	^
v	System PLL [7 - 0]			
	SYS_PLL_CFG [1 - 0]	0b00 - PLL is killed in 'Inter		
	SYS_PLL_RAT [6 - 2]	0b01110 - 14:1 🗸		
	PLATFORM_CLOCK	0b00100 - 4:1		
>	Memory Controller Complex PLL [23 - 8]	0b00110 - <mark>6</mark> :1		
~	Cluster Groups PLL [143 - 24]	0b01000 - 8:1		
	 Cluster Group A PLL [47 - 24] 	0b01010 - 10:1 0b01100 - 12:1		
	DDR2_PLL_CLOCK	0b01100 - 12:1 0b01110 - 14:1		
	CGA_PLL1_CFG [25 - 24]	0b10000 - 16:1		
	CGA_PLL1_RAT [31 - 26]	0b001111 - 15:1 (Async Mo		
	CGA_PLL1_CLOCK	1.500 GHz		
	CGA_PLL2_CFG [33 - 32]	0b00 - Default value		
	CGA_PLL2_RAT [39 - 34]	0b010100 - 20:1 (Async Mo		
	CGA_PLL2_CLOCK	2.000 GHz		
	Cluster Group B PLL [127 - 48]			
	SYS_PLL_SPD [128 - 128]	0b0 - Default value		

The figure below shows an example of changing a field value by using an advanced GUI.

roperties Import Validation									✓ 🗖 🛛
C5 PLL SEL [163 - 160]	SRDS_PRTCL_S1				SER	DES1			
C5_PLL_SEL [105 - 100]		Lane 0	Lane 1	Lane 2	Lane 3	Lane 4	Lane 5	Lane 6	Lane 7
C6_PLL_SEL [167 - 164]	Ox0	off							
C6_PLL_CLOCK	• 0x1		PCI (16/5/				PCI (16/5	le.2 /2.5G)	
C7_PLL_SEL [171 - 168] C7_PLL_CLOCK	O 0x2	SGMII.3 (1.25G)	SGMII.4 (1.25G)	SGMII.5 (1.25G)	SGMII.6 (1.25G)		PC (16/5	le.2 /2.5G)	
C8_PLL_SEL [175 - 172] Clocking Configuration [228 - 176	O 0x3	USXGMII/XFI.3 (10.3125G)	USXGMII/XFI.4 (10.3125G)	USXGMII/XFI.5 (10.3125G)	USXGMII/XFI.6 (10.3125G)		PC	le.2 /2.5G)	
 Required Customer Configuration Boot Configuration [291 - 231] 	O 0x4	SGMII.3 (1.25G)	SGMII.4 (1.25G)	SGMII.5 (1.25G)	SGMII.6 (1.25G)	SGMII.7 (1.25G)	SGMII.8 (1.25G)	SGMII.9 (1.25G)	SGMII.10 (1.25G)
System Clock Frequency [301 - 29 Pin Multiplexing Configuration [8	O 0x5		PCI (16/5,			USXGMII/XFI.7 (10.3125G)	USXGMII/XFI.8 (10.3125G)	USXGMII/XFI.9 (10.3125G)	USXGMII/XFI.10 (10.3125G)
 SerDes PLL and Protocol Configur SerDes PLL Configuration [911 	O 0x6	USXGMII/XFI.3 (10.3125G)	USXGMII/XFI.4 (10.3125G)	SGMII.5 (1.25G)	SGMII.6 (1.25G)	SGMII.7 (1.25G)	SGMII.8 (1.25G)	SGMII.9 (1.25G)	SGMII.10 (1.25G)
 SerDes Protocol Selection [931 SRDS_PRTCL_S1 [916 - 912] 	O 0x7	USXGMII/XFI.3 (10.3125G)	USXGMII/XFI.4 (10.3125G)	USXGMII/XFI.5 (10.3125G)	USXGMII/XFI.6 (10.3125G)	SGMII.7 (1.25G)	SGMII.8 (1.25G)	SGMII.9 (1.25G)	SGMII.10 (1.25G)
SRDS_PRTCL_S2 [921 - 917] SRDS_PRTCL_S3 [926 - 922]	O 0x8	USXGMII/XFI.3 (10.3125G)	USXGMII/XFI.4 (10.3125G)	USXGMII/XFI.5 (10.3125G)	USXGMII/XFI.6 (10.3125G)	USXGMII/XFI.7 (10.3125G)	USXGMII/XFI.8 (10.3125G)	USXGMII/XFI.9 (10.3125G)	USXGMII/XFI.10 (10.3125G)
SRDS_REFCLKF_DIS_S1 [927 - SRDS_REFCLKF_DIS_S2 [928 -		PCIe.1	SGMII.4	SGMII.5	SGMII.6	PCIe.2	SGMII.8	SGMII.9	SGMII.10
SRDS_REFCLKF_DIS_S3 [929 -	Save Restore								

4.2 Specify custom values

You can observe that some fields on the **Properties** page do not show a list of possible values and some other fields show a list of values where few values are missing. This is due to the reason that by default, the PBL tool restricts field values to the ranges specified in the SoC reference manual. The PBL tool also validates entered values against known constraints and generates errors in the Component Inspector view and Problems view when constraints are violated.

Both these features can be turned off by enabling the **Skip error checking** option available on the **Enable/disable error checking** toolbar menu. This allows you to set any RCW field to any value, even if such a configuration is likely or certain to cause the SoC fail to come out of reset or function improperly.

To specify a custom value for an RCW field, follow these steps:

1. In the Component Inspector view, click **Enable/disable error checking** and choose **Skip error checking**, as shown in the figure below.

Properties Import Validation			
Name	Value	Details	Check for errors
RCW Source	NOR Flash 16-bit Port size		Skip error checking
 			
• ⁽⁴⁾ PLL Configuration [175 - 0]			
 System PLL [7 - 0] 			
SYS_PLL_CFG [1 - 0]	0b00 - PLL is killed in 'Inter		
SYS_PLL_RAT [6 - 2]	0b01110 - 14:1		

2. Select the property you want to specify a custom value for, add the new value in the **Value** column for the property (see figure below), and press **Enter**.

roperties Import Validation			 🖌 🗖 🔚 🔇
lame	Value	Details	
RCW Source	NOR Flash 16-bi	t Port size	
Reset Configuration Word	[1023 -		
 PLL Configuration [175 	- 0]		
 System PLL [7 - 0] 			
SYS_PLL_CFG [1 - 0] 0b00 - PLL is kill	ed in 'Inter	
SYS_PLL_RAT [6 - 2] 10101	×	
PLATFORM_CLOCK	800.000 MHz		
> Memory Controller C	Complex		
> Cluster Groups PLL [143 - 24		
> Cluster PLL Selection	ı [175 - 1		
> Clocking Configuration	[228 - 1]		

The value added by you is marked as a custom value for the field and it is available for selection in the list of values, as shown in the figure below.

Properties Import Validation		✓ 🗖 🗒 %
Name	Value	Details
RCW Source	NOR Flash 16-bit Port size	
🗸 🐵 Reset Configuration Word [1023 - 0]		
🗸 🝈 PLL Configuration [175 - 0]		
🗸 🝈 System PLL [7 - 0]		
SYS_PLL_CFG [1 - 0]	0b00 - PLL is killed in 'Inter	
SYS_PLL_RAT [6 - 2]	0b10101 - Custom value 🗸 🗸	Property SYS_PLL_RAT is set to a custom value.
PLATFORM_CLOCK	0b00100 - 4:1	
> Memory Controller Complex PLL [23 - 8]	0b00110 - 6:1	
Cluster Groups PLL [143 - 24]	0b01000 - 8:1 0b01010 - 10:1	
> Cluster PLL Selection [175 - 144]	0b01010 - 10:1	
Clocking Configuration [228 - 176]	0b01110 - 14:1	
Required Customer Configuration [230 - 229]	0b10000 - 16:1	
Boot Configuration [291 - 231]	0b10101 - Custom value	
System Clock Frequency [301 - 292]		

4.3 Display and set reserved fields

After enabling the **Skip error checking** option, you can edit the values for the reserved bit fields, apart from the non-reserved bit fields. However, before editing reserved bit field values, you need to make reserved bit fields visible by choosing **Display all fields** from the **View Mode** toolbar menu, as shown in the figure below.

Properties Import Validation			🖬 🖌 🗖 🖾 🖇
Name	Value	Details	Display only non-reserved fields
RCW Source	NOR Flash 16-bit Port size		Display all fields
🕗 💩 Reset Configuration Word [1023 - 0]			
 			
🗸 🝈 System PLL [7 - 0]			
SYS_PLL_CFG [1 - 0]	0b00 - PLL is killed in 'Inter		
SYS_PLL_RAT [6 - 2] SYS_PLL_RAT [7 - 2]	0b10101 - Custom value	Property S	SYS_PLL_RAT is set to a custom value.
PLATFORM_CLOCK	2.100 GHz		
RESERVED_7 [7 - 7]	0b0 - Default value		

4.4 Add PBI commands to a PBL image

You can add the PBI commands to a PBL image by using the **Pbi Data** property under the **PBI Data** category on the **Properties** page of the **Component Inspector** view. Perform the following steps to add the PBI commands:

1. Expand the **PBI Data** property category on the **Properties** page of the **Component Inspector** view and click anywhere within any cell of the **Pbi Data** row, as shown in the figure below.

roperties Import Validation			🖌 🗖 🗒 🖗
lame	Value	Details	
RCW Source	NOR Flash 16-b	it P	
Reset Configuration Word [1023 - 0]			
> PLL Configuration [175 - 0]			
> Clocking Configuration [228 - 176]			
> Required Customer Configuration [230 -	229]		
> Boot Configuration [291 - 231]			
System Clock Frequency [301 - 292]			
> Pin Multiplexing Configuration [895 - 302	2]		
> SerDes PLL and Protocol Configuration [1	023 -		
💧 PBI Data			
💩 Pbi Data	No Data	Click to display custom editor for this property	
CRC in final Stop command	yes	Warning Generating CRC in final stop PBI command is no.	
PBL Data			

The **PBI Data Input** editor opens, as shown in the figure below.

roperties Import Validation		🖌 🗖 🔚 🖇
Name	PBI Data Input	
RCW Source		
 Reset Configuration Word [1023 - 0 	Select PBI command CCSR Write ~	
> PLL Configuration [175 - 0]	Command parameters	
> Clocking Configuration [228 - 176	SYS ADDR (0x) 0000000	
> Required Customer Configuration		
> Boot Configuration [291 - 231]	CCSR_DATA (0x) 0000000	
> System Clock Frequency [301 - 29	Byte Count 4 V	
> Pin Multiplexing Configuration [8		
> SerDes PLL and Protocol Configur	Add Command	
🖌 🙆 PBI Data	• Add command	
🍈 Pbi Data	Added PBI Commands:	🗙 🔆 🗘 🖓 🖬 🖥
CRC in final Stop command		
> PBL Data		
	Save Restore	

2. Choose the appropriate PBI command from the Select PBI command menu, as shown in the figure below.

PBI Data Input			
Select PBI command	CCSR Write ~		
CCSR_DATA (0x) 00	AltConfig Write Block Copy Load Alternate Configuration Window Load Condition Load Security Header		
Added PBI Commands:	Load Boot 1 CSF Header Pointer Poll (short) Poll (long) Wait Jump Jump Conditional Stop	ify Command	🛠 🏠 🕀 🖻 🛛
gure 14. Choosing a P	BI command		

3. Edit command parameters in the **Command parameters** group and click the **Add Command** button. The PBI command is added in the **Added PBI Commands** pane, as shown in the figure below.

Properties Import Validation	✓ □ 🗟 🖇
Name RCW Source Reset Configuration Word [1023 - 0 > PLL Configuration [175 - 0] > Clocking Configuration [228 - 17(> Required Customer Configuratior > Boot Configuration [291 - 231] > System Clock Frequency [301 - 25 > Pin Multiplexing Configuration [8 > SerDes PLL and Protocol Configur • PBI Data • PBI Data • CRC in final Stop command > PBL Data	PBI Data Input Select PBI command Command parameters SYS_ADDR (0x) 1e00200 CCSR_DATA (0x) Question 4 Add Command Modify Command Add Command CCSR 4-byte Write to 0x01e00200, data=0x12345678
	Save Restore

You can view the PBI commands in two modes: disassembly view and raw data view. To switch between the two modes, click the rightmost button on the toolbar of the **Added PBI Commands** pane, as shown in the following figures.

Added PBI Commands:	×	*	Ŷ	Ŷ	1
CCSR 4-byte Write to 0x01e00200, data=0x12345678					
Figure 16. Disassembly view					
Added PBI Commands:					
31E0020012345678					^
Figure 17. Raw data view					

4. Click the Save button to add the PBI commands to the PBL image.

Select PBI command	CCSP Write	\sim	
Select PBI command	CCSK WITTE	~	
Command parameter	S		
SYS_ADDR (0x) 1e	00200		
CCSR_DATA (0x) 12	345678		
Byte Count 4	~		
		-214 116 0	
	👎 Add Command	Modify Command	
Added PBI Commands		Modify Command	🗙 🦗 수 🕂 🇗
	:		X 🔆 t 🤣 🗹 🖥
Added PBI Commands CCSR 4-byte Write to			X 🔆 û 4 🗹 🖬
	:		X 🔆 t 🤣 🗹 🖥
	:		🗙 🔆 û 🤄 🗹 🖬
	:		X 🔆 û 4 🗹 🖬
	:		🗶 🔆 û 🤑 🗹 🗟
	:		X 🔆 û 4 🗹 🖬
	:		🗙 🔆 û 4 🗹 🖬
	:		🗶 🔆 û 🤄 🗹 🗟
	:		🗶 🔆 û

NOTE

Depending on its parameters, a PBI command may need to be split into several PBI commands. The PBL tool does it automatically and informs the user about the split.

4.5 Import a PBL configuration from a file

To import a PBL configuration for an existing PBL component, perform these steps:

- 1. Click the Import tab in the Component Inspector view. The Import page appears.
- 2. Choose a PBL file by clicking the **Load from file** button. The file format of the chosen file is automatically detected and its content is displayed in the Rich Text Format in an editor available in the **Input data** group, as shown in the figure below.

roperties Impo	ort V	alidati	on														<u> </u>	/ 🗖 🖪 🕯
Get PBL data fr	om file	or fro	m ta	rget -	data	a and	d dat	a foi	mat	will	be in	serte	ed in	to th	e ing	out area below.		
Load from file		and fr		ract														
Load from the		ead fr	omita	irget.														
Input data —																		
Format: xx	d Obje	ect Dur	nn					~										
			-															
Endianness: Li	ttle En	dian 🔿	~															
Additional Dat	a. ¢/r	roject	Vnbl	tag	alon	a hin	*	2										
								•										
00000000:							80	b0	30	30	40	40	00			U.U0000.00		^
00000010:				00			00	00	00		00			00				
00000020:					00		00	80	11	00	01			00		· · · · · · · · · · · · · · · · · · ·		
00000030:				00					0c		00			00				
00000040:		00 00	00	00	00	00	00	00	00	00								
00000050:		00 00	00	00	00	00	00	00	00	00				00	00			
00000060:	00 0	00 00	00	00	00	00	00	00	00	00	00	00	00	00	00			
00000070:		0 02	00	00	00	00	00	00	00	2a				00		.p*A		
00000080:		00 00	00	00	00	00	00	00	00	00			02	e0		1		
00000090:		6 34				e0	31	00	00	00			04			xV411		
000000a0:	00 0	0 10	30	9c	09	eb	31	80	28	50	00					01.(P1		
000000b0:	80 2	28 50	00	00	13	eb	31	20	4e	10	80	10	13	eb	31	.(P1.N1		~
<		10			-	~	-	-	-	-	-							>
												In	port					

- 3. Edit the PBL file in the editor, as needed.
- 4. Click the Import button to import the new PBL configuration.
- 5. Switch to the Properties tab to view or edit the imported PBL configuration.

Using the steps provided in this section, you can import files having the following file formats:

- XXD Object Dump
- U-Boot Flash Dump
- Hex String
- U-Boot CCRS Startup Dump (RCW only)
- Hex String (RCW only)
- Text Table (RCW only)

NOTE

By default, the import operation tries to convert the memory dump into the XXD Object Dump format and displays it in the Rich Text Format.

4.6 Import a PBL configuration from target

To import a PBL configuration for an existing PBL component, perform these steps:

- 1. Click the Import tab in the Component Inspector view. The Import page appears.
- 2. Click the **Read from target** button. The file format of the chosen file is automatically set to **Import From Target** and it starts reading the RCW information from the current target; when the data is read, it will be displayed in the **Input data** group, as shown in the figure below.

Before choosing to read PBL information from target, ensure that you have activated the right target in the **Target** connections group of the **Connections View**.

Project Explorer 🛤 Connections Vi 🕺 📱 🗖	🗞 Component Inspector - PBL1 🕴 🗞 Components Library	Basic Advanced	<u> </u>	8 -	
	Properties Import Validation		√ [3 6	1 🗞
Target connections	Get PBL data from file or from target - data and data format will be inserted into the input area below.				
🕐 🗟 🗙 🚺 🖉 🖉	Load from file Read from target				
Processor Probe T Probe Address	Input data				
LX2160A cwtap 10.171	Format: Import From Target V				
LX2160A cwtap 10.771 active connection	Additional Data: \${project}/pbl_tag_along.bin 🔀 RCWSR registers were successfully read from the target and imported!				^
	RCWSR1:0x58777738 RCWSR2:0x24580058 PCWSP3:0x20000000				~
	< Import			>	
igure 20. Read PBL configurati				2	-

- 3. Click the Import button to import the new PBL configuration.
- 4. Switch to the Properties tab to view or edit the imported PBL configuration.

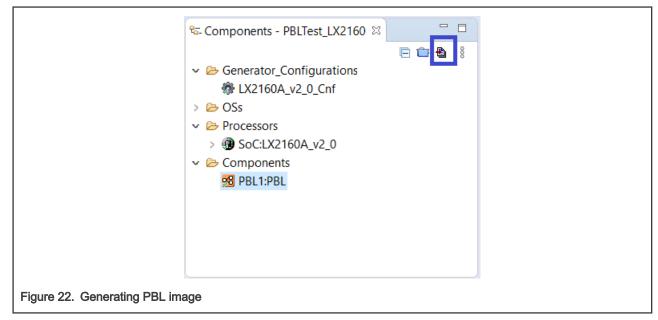
4.7 Generate a PBL image

To generate a PBL image from the PBL configuration, perform these steps:

- 1. Select the **Output Format** toolbar button.
- 2. Choose a file format for the PBL image.

roperties Import Validation			✓ □ 📓
Name	Value	Details	Binary
RCW Source	NOR Flash 16-bit Port size		xxd Object Dump
Reset Configuration Word [1023 - 0]			U-Boot Commands
 PLL Configuration [175 - 0] 			Hex String
> System PLL [7 - 0]			Hex String (RCW only)
 Memory Controller Complex PLL [23 - 8] 			Text Table (RCW only)
DDR_REF_CLOCK	100.000 MHz		
MEM_PLL_CFG [9 - 8]	0b00 - Setting selects DDR		
MEM_PLL_RAT [15 - 10]	0b000110 - 6:1 (Async Mod		
DDR1_PLL_CLOCK	600.000 MT/s		

3. Click the Generate Processor Expert Code icon in the Components view to generate the PBL image.



Using the steps provided in this section, you can generate PBL images with the following file formats:

- Binary
- XXD Object Dump
- U-Boot Commands
- · Hex String
- · Hex String (RCW only)
- Text Table (RCW only)

4.8 Automatic PBL validation and errata support

Each time you change the PBL configuration, the PBL tool performs a check against known constraints and issues. If the configuration is found invalid, then the error or warning messages are displayed for the problematic RCW fields on the **Properties** page in the **Component Inspector** view, as shown in the figure below. The error or warning messages are also displayed with additional details in the **Problems** view, as shown in the figure below.

You can use **Skip error checking** mode to suppress critical errors; they will be marked as warnings and you will be able to generate the configuration.

The figure below illustrates two situations: an error caused by a field set to a custom value and a warning caused by an errata check.

igure 23. PBL validatio)
	al Stop PBI command is not supported fo	or Non-E parts (i.e. security and ci	yprograpny block not available) due to nardware errata.	PBLIEST_LX2160	
Warnings (1 item)	al stan DDI command is not supported f	or Non E parts (i.e. cocurity and c	yptography block not available) due to hardware errata.	PBLTest LX2160	
Property MEM2_PLL_RAT is set	to a custom value.			PBLTest_LX2160	
🗸 🥹 Errors (1 item)					
Description		^		Resource	Path
error, 1 warning, 0 others					
Problems 23 🗞 PBL Configuration Re	eaisters			8	8 - 0
😼 PBL1:PBL	CRC in final Stop command PBL Data	yes	Warning Generating CRC in final stop PBI command is not s	supported for Non-E parts (i.	
Components	Pbi Data	No Data	Click to display custom editor for this property		_
> SoC:LX2160A_v2_0	v 🚯 PBI Data				
Processors	SerDes PLL and Protocol Co	onfigu			
🗁 OSs	> Pin Multiplexing Configurat	-			
LX2160A_v2_0_Cnf	> System Clock Frequency [30]	01 - 2			
Generator_Configurations	Boot Configuration [291 - 2				
E 💼 월 🖇	Required Customer Configu				
- Components - PBLTes 🛛 🗖 🗖	 Clocking Configuration [22] 				
	Cluster Groups PLL [143				
	DDR2_PLL_CLOCK	6.300 GT/s			
	MEM2_PLL_RAT [23 - 1]		Property MEM2_PLL_RAT is set to a custom value.		
	MEM2 PLL CFG [17 - 1			Skip error check	ing
PBLTest_LX2160	Name	Value	Details	Check for errors	
🖻 🕏 🏹 🛸 🕴	Properties Import Validation			4	- 18 %

4.9 PBL validation tool

The PBL validation tool can be used for RCW validation, after a PBL configuration is created using the PBL configuration tool.

The Component Inspector view displays a page, **Validation**, which represents the GUI of the PBL validation tool. To use the PBL validation tool, select the **Validation** tab of the **Component Inspector** view, as shown in the figure below.

If you click the **Write Reset Configuration Word** button, it will override the RCW on target and will perform a target reset. If the processor core is working correctly with the written RCW and it does not return any error, then the test will pass.

Properties Import Validation Write Reset Configuration Word	🖌 🗖 🗒 🖗
Write Reset Configuration Word	
while Reset conliguration word	
Reset Configuration Word:	
RCWSR1:0x58777738	~
RCWSR2:0x24580058	
RCWSR3:0x00000000	
RCWSR4:0x00000000	
RCWSR5:0x00000000	
RCWSR6:0x0c010000	
RCWSR7:0x00000000	
RCWSR8:0x00000000	
RCWSR9:0x000001a0	
RCWSR10:0x00002580	
RCWSR11:0x00000000	
RCWSR12:0x00000096	
RCWSR13:0x00000000	
RCWSR14:0x0000000	
RCWSR15:0x0000000	
RCWSR16:0x0000000	
RCWSR17:0x0000000	
RCWSR180x0000000	
Test Result	
Test passed!	

4.10 Synchronize PBL with other IP blocks

You can synchronize the PBL component with other intellectual property (IP) blocks, such as SerDes or DDR, if the corresponding component is available in the current project.

To synchronize the PBL component with a SerDes component, perform these steps:

- 1. Ensure that a SerDes block component is available in the current project.
- Double-click a SerDes component grouped under the SerDes block component in the Components view. The properties
 of the SerDes component are displayed on the SerDes Configuration and Validation page in the Component Inspector
 view.
- 3. Click the **Apply the configuration to PBL component** button at the top-left corner (second button) of the **SerDes Configuration and Validation** page to synchronize the PBL component with the SerDes component, as shown in the figure below.

25	onfiguration and	d Validatio	'n														
Арр	ly the configura	ation to PB	L componer	nt. G	Lar	ne F	La	ne E	Lar	ne D	Lan	e C	Lar	ne B	La	ne A	
T LL	SD1_T 9	SD1	SD1_T	SD1_	SD1_T	SD1	SD1_T	. SD1	SD1_T	SD1	SD1_T	SD1	SD1_T	SD1	SD1_T	SD1	
					PCle1								Cle2				
	0	0	0	0	(2.5)	٩	0	٩	۵	٩	0		2.5)	٩	0	0	
	420	450	424	424	420	420	424	424	450	420	(21)	424	4 20	420	420	420	
PLL F					\checkmark	×	×	×									
PLL S	×	×	1	1					~	~	1	1	~	×	1	~	
_	vert data				Invert data						🗌 Boost						
▼ Ec	qualization				★ Electrical	idle					Gaink2				_		
Тур	e	2 Leve	els	\sim	Threshold	Disable	e LOS		\sim		Source	Use rxeq ad	daption deriv	/ed gaink2	~		
	Cursor sign	1 ~			Enter idle fi	Iter Bypass	Unexpected	Entrance int	o Idle 🛛 🗸		Value	0 ~					
Pre	Cursor ratio	No ec	qualization	\sim	Exit idle filt	er Force E	xit AFTER M	lin Time in Id	le Data stop	ped 🗸 🗸	Gaink3						
		1 ~									Source	Use rxeq ad	daption deriv	ved gaink3	~		
Pre	tCursor sign			\sim							Value	0 ~					
Pre Pos	tCursor sign tCursor ratio	1.14		~													

4. Select the PBL component in the **Components** view and verify the SerDes fields on the **Properties** page of the **Component Inspector** view, as shown in the figure below.

Properties Import Validation									🖌 🗖 🛛	
Name	SRDS PRTCL S1	S1 SERDES1								
RCW Source		Lane 0	Lane 1	Lane 2	Lane 3	Lane 4	Lane 5	Lane 6	Lane 7	
 Reset Configuration Word [1023 - 0 PLL Configuration [175 - 0] 	O 0x0	off	off	off	off	off	off	off	off	
Clocking Configuration [228 - 176	• 0x1	PCle.1 PCle.2 (16/5/2.5G) (16/5/2.5G)								
 Required Customer Configuration Boot Configuration [291 - 231] 	O 0x2	SGMII.3 (1.25G)	SGMII.4 (1.25G)	SGMII.5 (1.25G)	SGMII.6 (1.25G)			le.2 (/2.5G)		
 System Clock Frequency [301 - 29 Pin Multiplexing Configuration [8 	O 0x3	USXGMII/XFI.3 (10.3125G)	USXGMII/XFI.4 (10.3125G)	USXGMII/XFI.5 (10.3125G)	USXGMII/XFI.6 (10.3125G)			le.2 (/2.5G)		
 SerDes PLL and Protocol Configur SerDes PLL Configuration [911 	O 0x4	SGMII.3 (1.25G)	SGMII.4 (1.25G)	SGMII.5 (1.25G)	SGMII.6 (1.25G)	SGMII.7 (1.25G)	SGMII.8 (1.25G)	SGMII.9 (1.25G)	SGMII.10 (1.25G)	
 SerDes Protocol Selection [931 SRDS_PRTCL_S1 [916 - 912] 	○ 0x5	(11230)	PCI (16/5/	e.1	(11250)	USXGMII/XFI.7 (10.3125G)	USXGMII/XFI.8 (10.3125G)	USXGMII/XFI.9 (10.3125G)	USXGMII/XFI.10 (10.3125G)	
SRDS_PRTCL_S2 [921 - 917] SRDS_PRTCL_S3 [926 - 922] SRDS_REFCLKF_DIS_S1 [927 - >	Save Restore		1	I			I			

4.11 View RCW status registers

To have an overview of the RCW status registers, perform these steps:

1. Choose **PBL Configuration Registers** from the toolbar. The **PBL Configuration Registers** view appears, displaying the details of the RCW status registers, as shown in the figure below.

E	Properties Import Validation 🗸 🗖 🔚 🦠			
	Name		Value PB	L Configuration Registe
	RCW Source		NOR Flash 16-bit Port size	
	 Reset Configuration 	Word [1023 - 0]		
Components - P 🛛 🗖 🗖	> PLL Configuration	n [175 - 0]		
	> Clocking Configu	Clocking Configuration [228 - 176]		
LX2160A v2 0 Cnf ^	> Required Custom	Required Customer Configuration [230 - 229]		
≥ OSs	> Boot Configuration	Boot Configuration [291 - 231]		
Processors	> System Clock Free	quency [301 - 292]		
> 🛞 SoC:LX2160A_v2_0		Configuration [895 - 302]		
Components	CauDae DI Laud D		4	>
Problems 🗞 PBL Configuration	-			- [
eg. name	Value	Address		
RCWSR1	58777738	0x1E00100		
SYS_PLL_CFG [1 - 0]	00			
SYS_PLL_RAT [6 - 2]	01110			
RESERVED_7 [7 - 7]	0			
MEM_PLL_CFG [9 - 8]	11			
MEM_PLL_RAT [15 - 10]	011101			
MEM2_PLL_CFG [17 - 16]	11			
MEM2_PLL_RAT [23 - 18] CGA_PLL1_CFG [25 - 24]	011101			
	00			

The **PBL Configuration Registers** view reflects any changes made in the PBL configuration. Also, from this view you can edit register bit field values or specify a certain value for the register; if you change the register value, then its bit fields and the corresponding properties will automatically be updated.

5 Advanced PBL operations

This section is divided into the following subsections:

- Add additional payload to a PBL image
- Endianness aspects

5.1 Add additional payload to a PBL image

You can add to a PBL image additional binary payload, such as U-Boot. This is useful to create boot images for the SPI/SD/NAND flash when PBL can be edited without decoupling it from the U-Boot binary. At code generation, the binary payload is automatically re-attached to the modified PBL.

To add additional binary data to the PBL image, perform these steps:

1. Expand the **PBL Data** property category on the **Properties** page of the **Component Inspector** view and click the **Additional Binary Data** property. The **Additional Binary Data** editor opens, as shown in the figure below.

Properties Import Validation		🖌 🗖 🗒 🖗
Name	 Additional Binary Data Specify file Location C:\Products\CW4NET_v2020.06\ws\u-boot-lx2160.bin Workspace File System Variables Placement Offset: 0x 300 Relative to RCW/PBI structure: O beginning O end 	

- 2. Specify a binary file to be added to the PBL image, in the Location field of the Specify file group.
- 3. Specify the offset and placement for the binary payload in the Placement group.
- 4. Click Save to apply your changes.

5.2 Endianness aspects

When you import data in the XXD Object Dump format, you can specify the endianness of the data. However, specifying the endianness is only useful when the data is organized into multibyte words. The endianness option is automatically set to the endianness of the chosen SoC, for example, little endian for the ARMv8-based SoCs.

6 PBL tool limitations

The PBL tool has some known limitations related to:

PBI commands

6.1 PBI commands

If the current PBL configuration has PBI commands defined and you import a new PBL image that does not have PBI commands, then the PBI commands are not preserved and you need to manually add them for the new PBL configuration.

A Revision history

The table below summarizes revisions to this document.

Table 1. Revision history

Revision	Date	Topic cross-reference	Change description
Rev. 1	11/2020		Updated entire document as per latest QCVS PBL tool

Table continues on the next page ...

Table 1. Revision history (continued)

Revision	Date	Topic cross-reference	Change description
			 Updated images throughout the document to show a latest supported device
Rev. 0	02/2016		Initial public release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, Freescale, the Freescale logo, CodeWarrior, Layerscape, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, and QUICC Engine are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Arm, CoreLink, CoreSight, Cortex, and TrustZone are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

[©] NXP B.V. 2016-2020.

All rights reserved.

For more information, please visit: http://www.nxp.com For sales office addresses, please send an email to: salesaddresses@nxp.com

> Date of release: 11/2020 Document identifier: AN5260

arm

