NXP Semiconductors
Application Note

Document Number: AN5260

PBL Configuration using QCVS

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

Contents

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 or imported from an existing reset configuration word (RCW) memory dump. Such memory dumps can be obtained using U-Boot.

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:

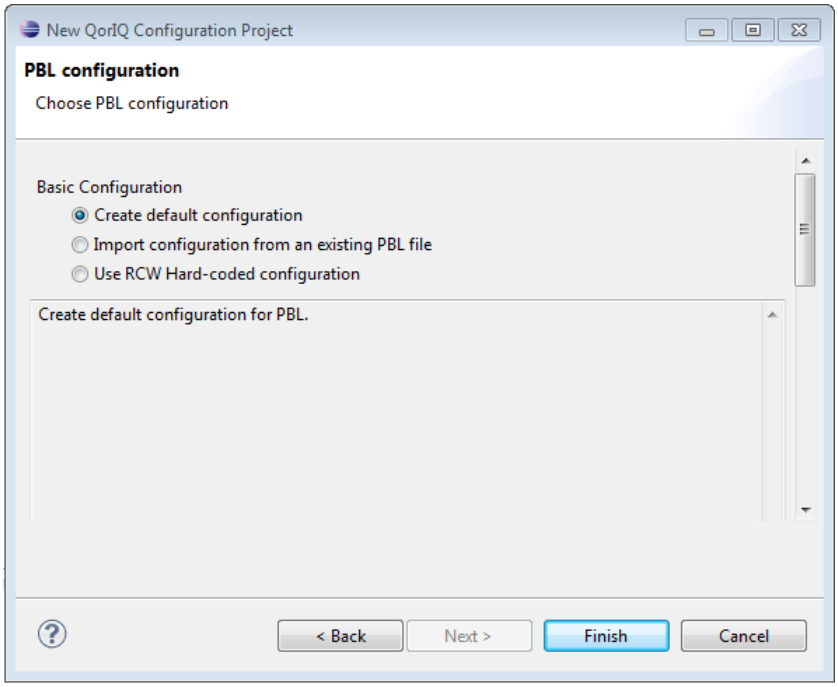
1. Open the QCVS Eclipse integrated development environment (IDE).
2. Choose **File > New > QorIQ Configuration Project** from the IDE menu bar. The **New QorIQ Configuration Project** wizard starts, displaying the **Create a QorIQ Configuration Project** page.
3. Specify the project name in the **Project 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
- Use RCW Hard-coded configuration

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 have a hard-coded RCW configuration to work on (see Use RCW Hard-coded configuration).
The figure below shows the **PBL configuration** page with the **Create default configuration** option selected.

Figure 1. Creating a default configuration

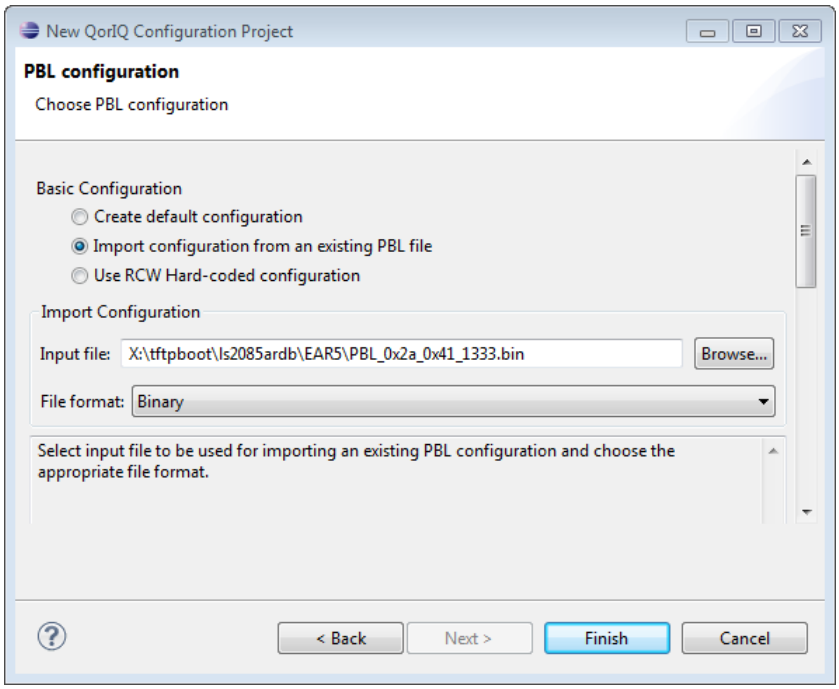


3.2. Import configuration from an existing PBL file

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.

Figure 2. Importing a configuration from an existing PBL file

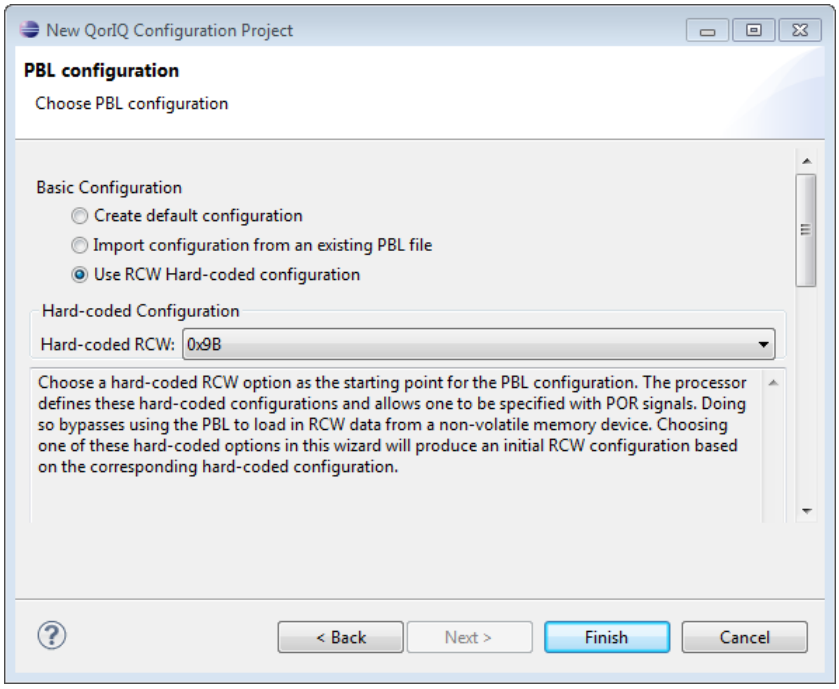


3.3. Use RCW Hard-coded configuration

A hard-coded 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 **Use RCW Hard-coded configuration** option to create a QorIQ configuration project with the PBL tool. Hard-coded RCWs have been tested with the QorIQ reference design boards (RDBs) and 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 **Use RCW Hard-coded configuration** option selected.

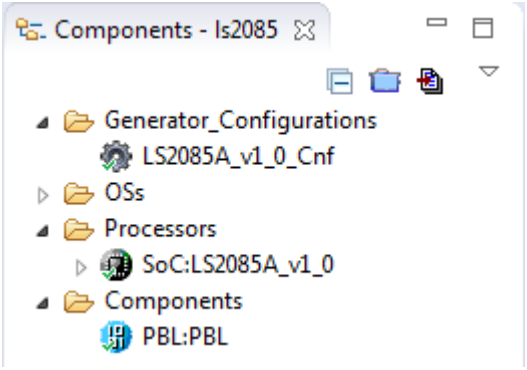
Figure 3. Using an RCW hard-coded configuration



4. Basic PBL operations

When you create a QorIQ 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.

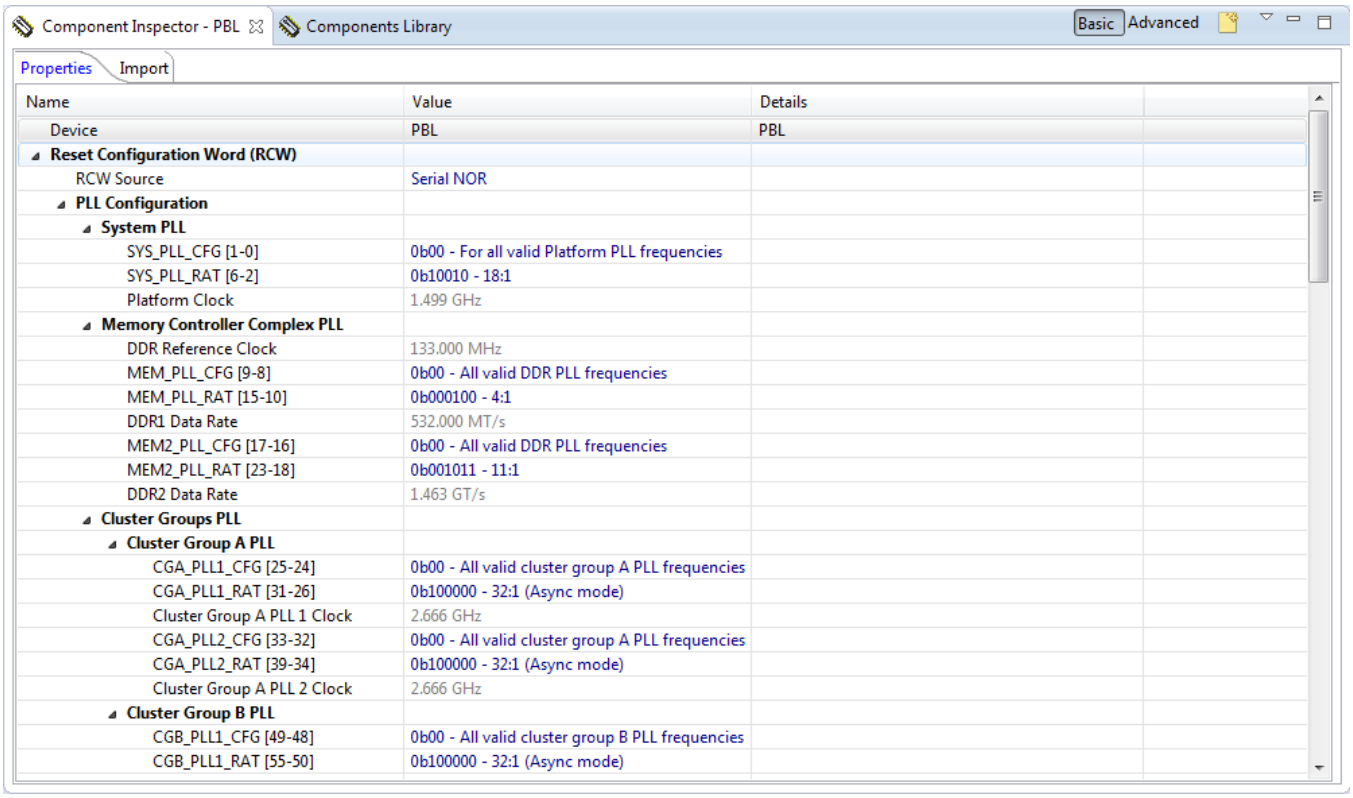
Figure 4. Components view



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.

Figure 5. Component Inspector view



As you can see in the figure above, the component properties are grouped under various groups and subgroups. All the settings and properties of the PBL component are sorted by the RCW position.

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

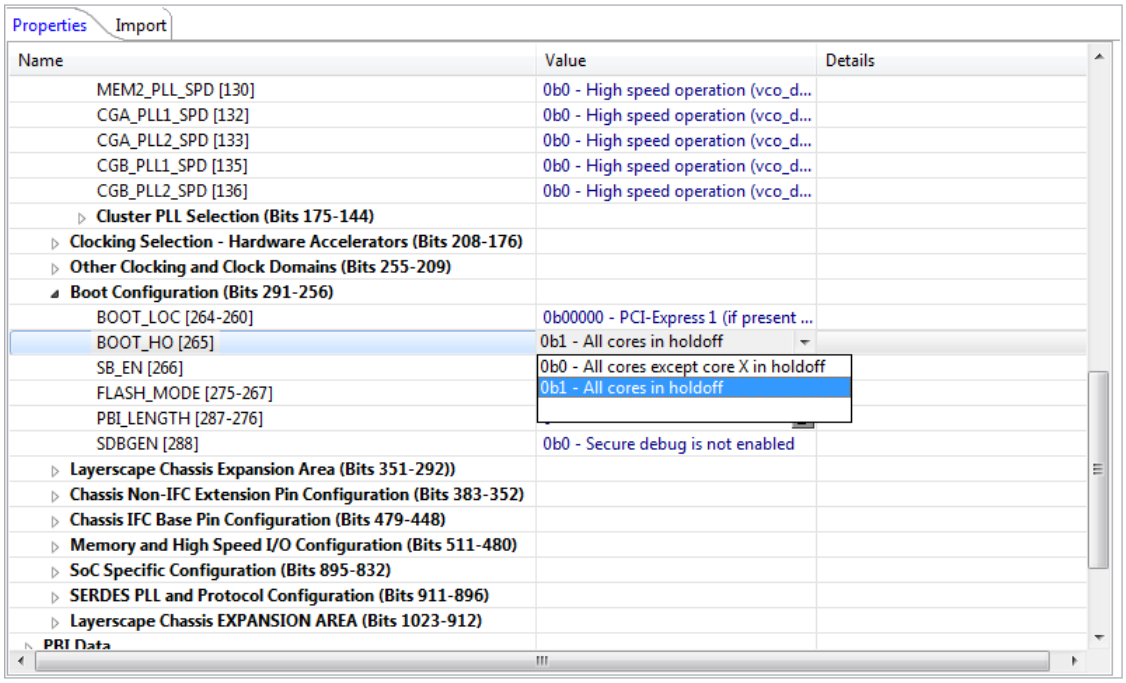
- Change RCW bit field values
- Add PBI commands to a PBL image
- Import a PBL configuration from a file
- Generate a PBL image
- Automatic PBL validation
- Synchronize PBL with other IP blocks
- View RCW status registers

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.

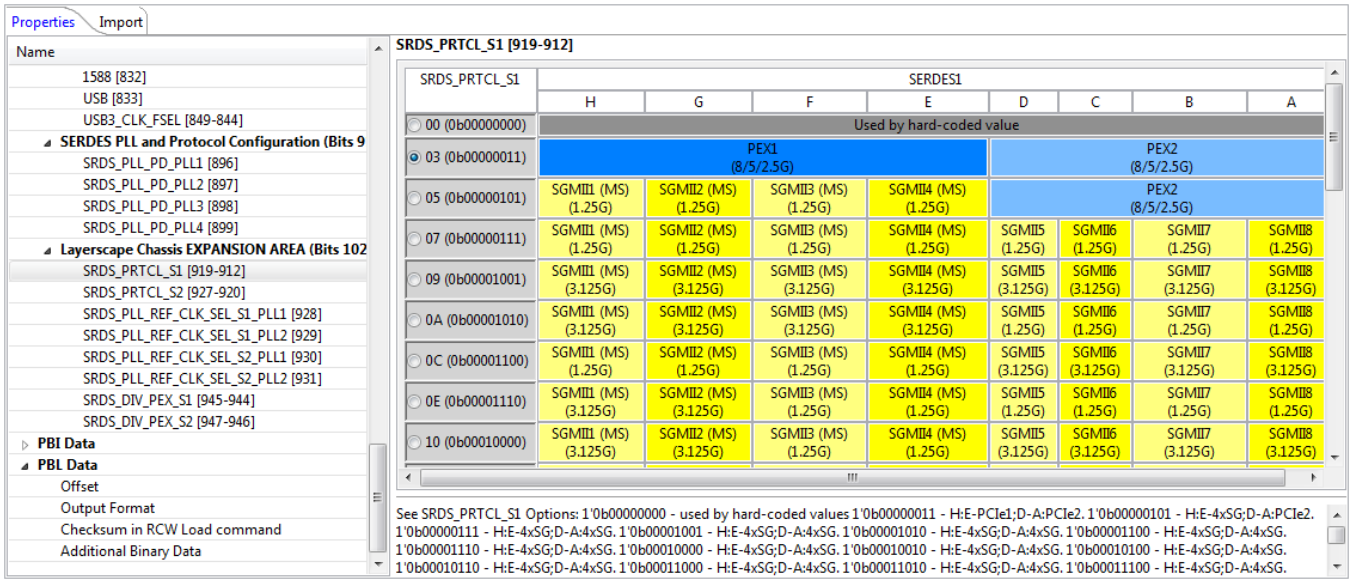
The figure below shows an example of changing a field value by choosing another value from a menu.

Figure 6. Changing a field value by choosing another value from a menu



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

Figure 7. Changing a field value by using an advanced GUI

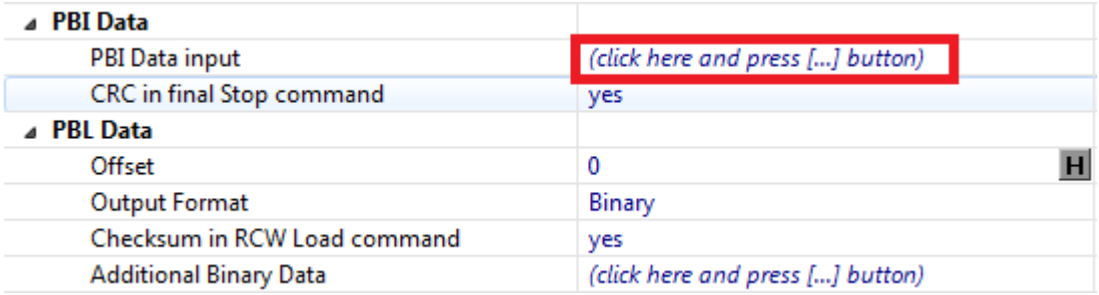


4.2. Add PBI commands to a PBL image

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

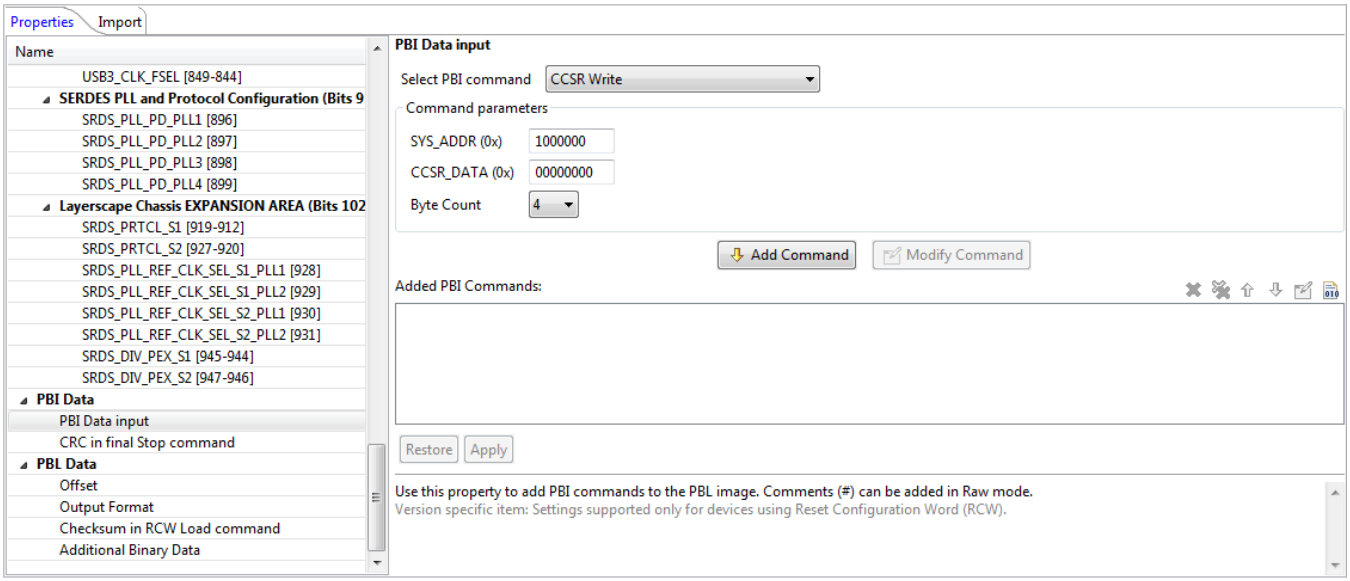
1. Select the **PBI Data input** property in the **Name** column and click the ellipsis (…) button in the **Value** column, as shown in the figure below.

Figure 8. Setting PBI Data input property



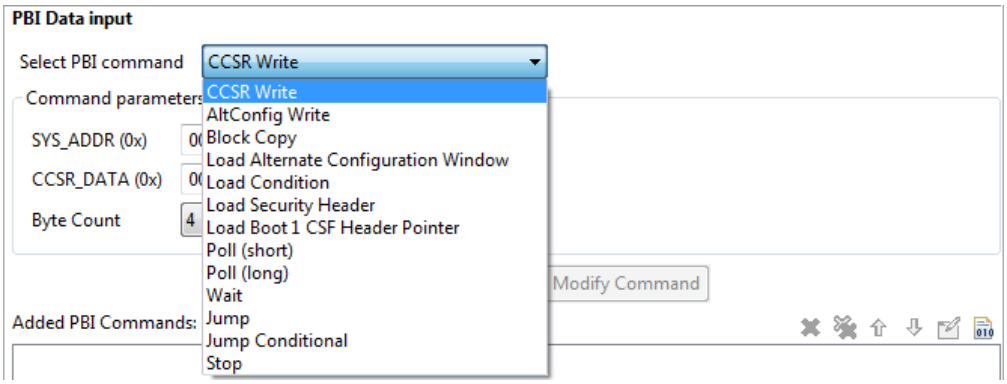
The **PBI Data input** editor opens, as shown in the figure below.

Figure 9. PBI Data input editor



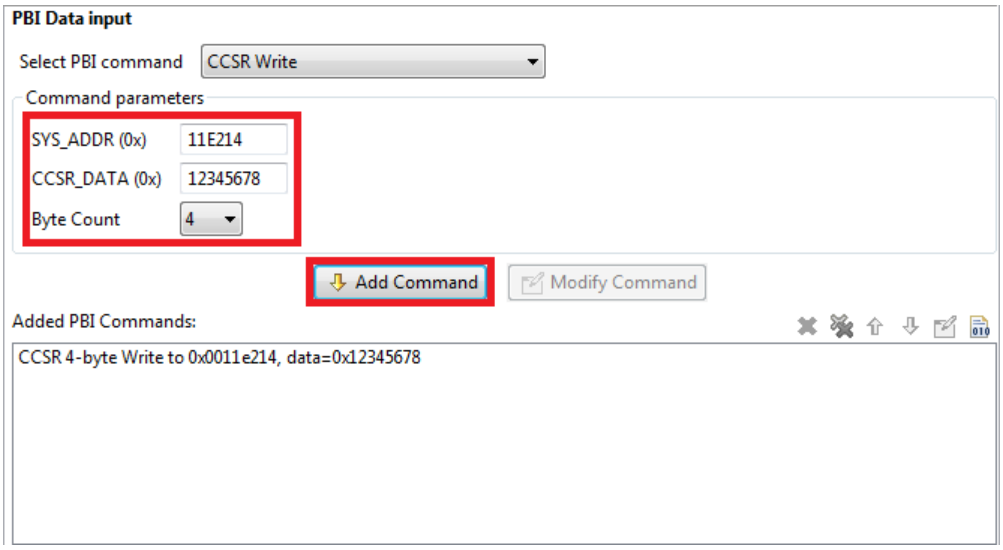
2. Choose the appropriate PBI command from the **Select PBI command** menu, as shown in the figure below.

Figure 10. Choosing a PBI 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.

Figure 11. Editing a PBI command



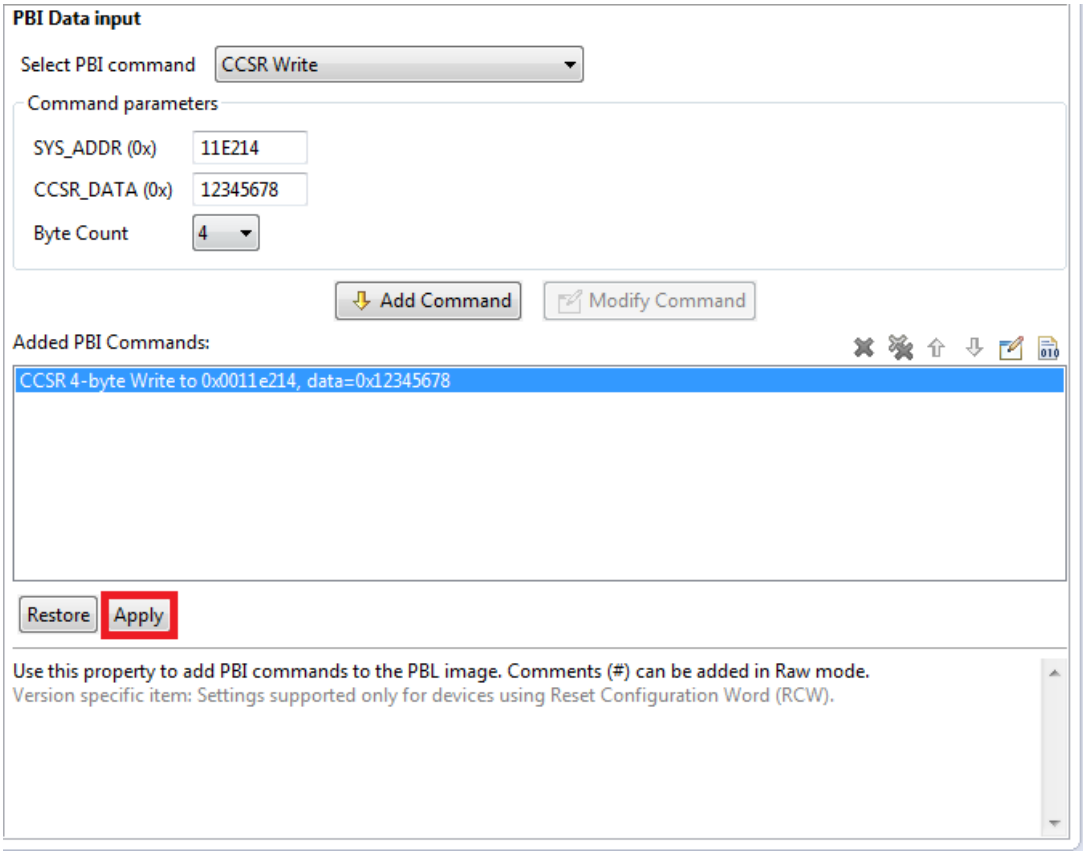
You can view the PBI commands in two modes: disassembly view and raw data view. To switch between the two modes, click the rightmost button on the toolbar of the **Added PBI Commands** pane, as shown in the following figures.

Figure 12. Disassembly view

Figure 13. Raw data view

4. Click the **Apply** button to add the PBI commands to the PBL image.

Figure 14. Adding PBI commands to a PBL image



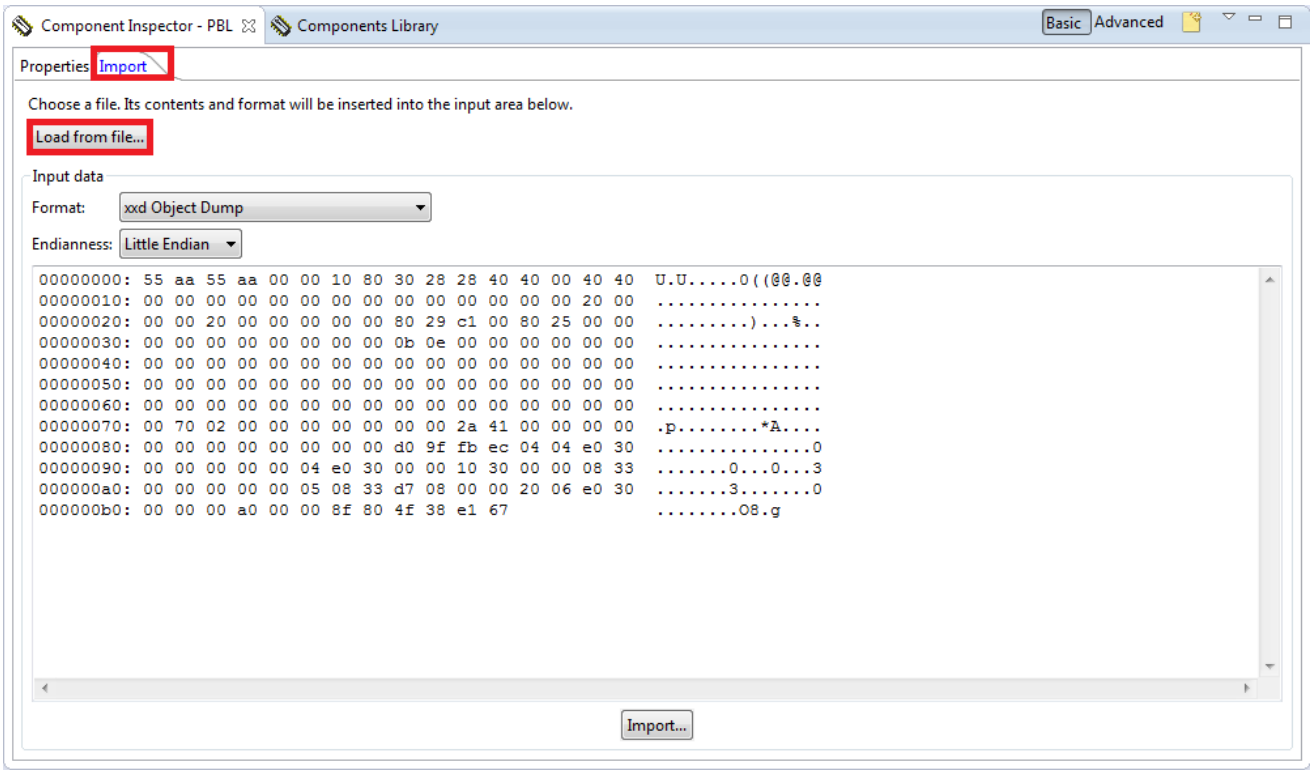
NOTE	Depending on its parameters, a PBI command may need to be split into several PBI commands. The PBL tool does it automatically and informs the user about the split.

4.3. 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 detected automatically 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.

Figure 15. Loading a PBL file



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
- S-record
- U-Boot Flash Dump
- Hex String
- U-Boot CCRS Startup Dump (RCW only)
- CW JTAG Config (RCW only)
- Hex String (RCW only)
- Test Table (RCW only)

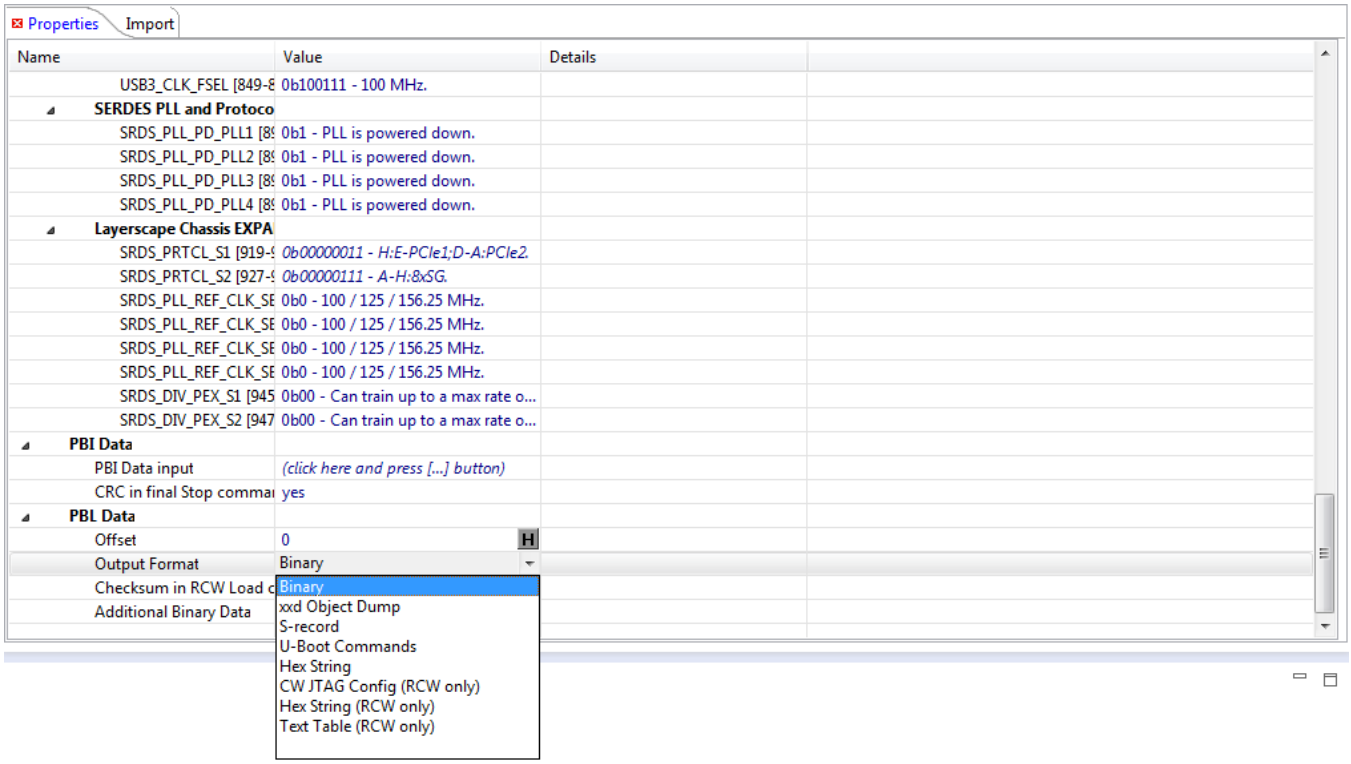
NOTE	By default, the import operation tries to convert the memory dump into the XXD Object Dump format and displays it in the Rich Text Format.

4.4. Generate a PBL image

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

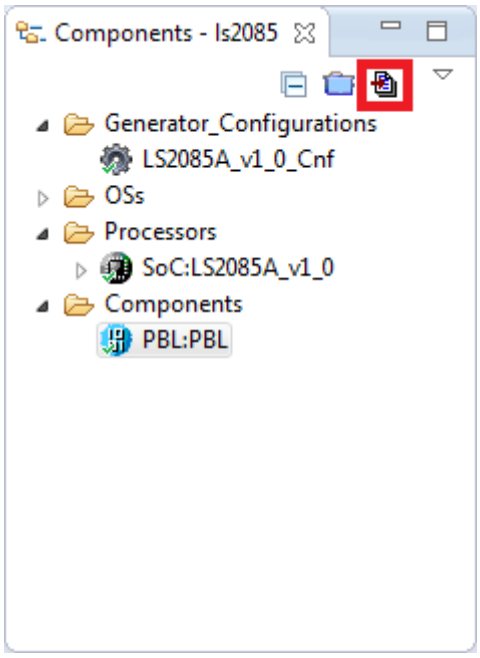
1. Select the **Output Format** property under the **PBL Data** group in the **Name** column on the **Properties** page of the **Component Inspector** view.
2. Click the corresponding cell in the **Value** column and choose a file format for the PBL image.

Figure 16. Choosing a file format for PBL file



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

Figure 17. Generating PBL image



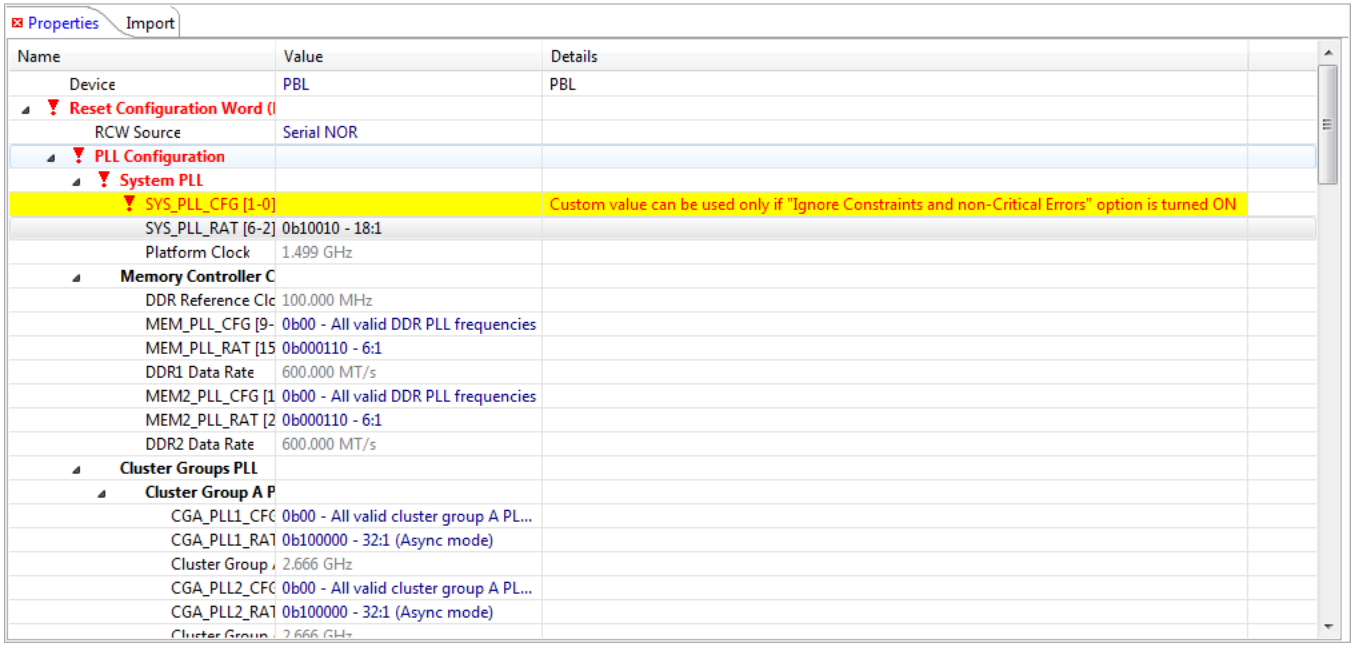
Using the steps provided in this section, you can generate PBL images with the following file formats:

- Binary
- XXD Object Dump
- S-record
- U-Boot Commands
- Hex String
- CW JTAG Config (RCW only)
- Hex String (RCW only)
- Text Table (RCW only)

4.5. Automatic PBL validation

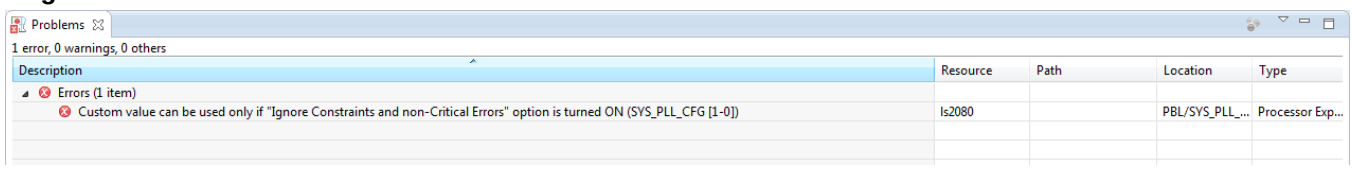
Each time you change the PBL configuration, the PBL tool performs a check against known constraints and issues. If the configuration is found invalid, then the error or warning messages are displayed for the problematic RCW fields on the **Properties** page in the **Component Inspector** view, as shown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

Figure 18. PBL validation



The error or warning messages are also displayed with additional details in the **Problems** view, as shown in the figure below.

Figure 19. Problems view



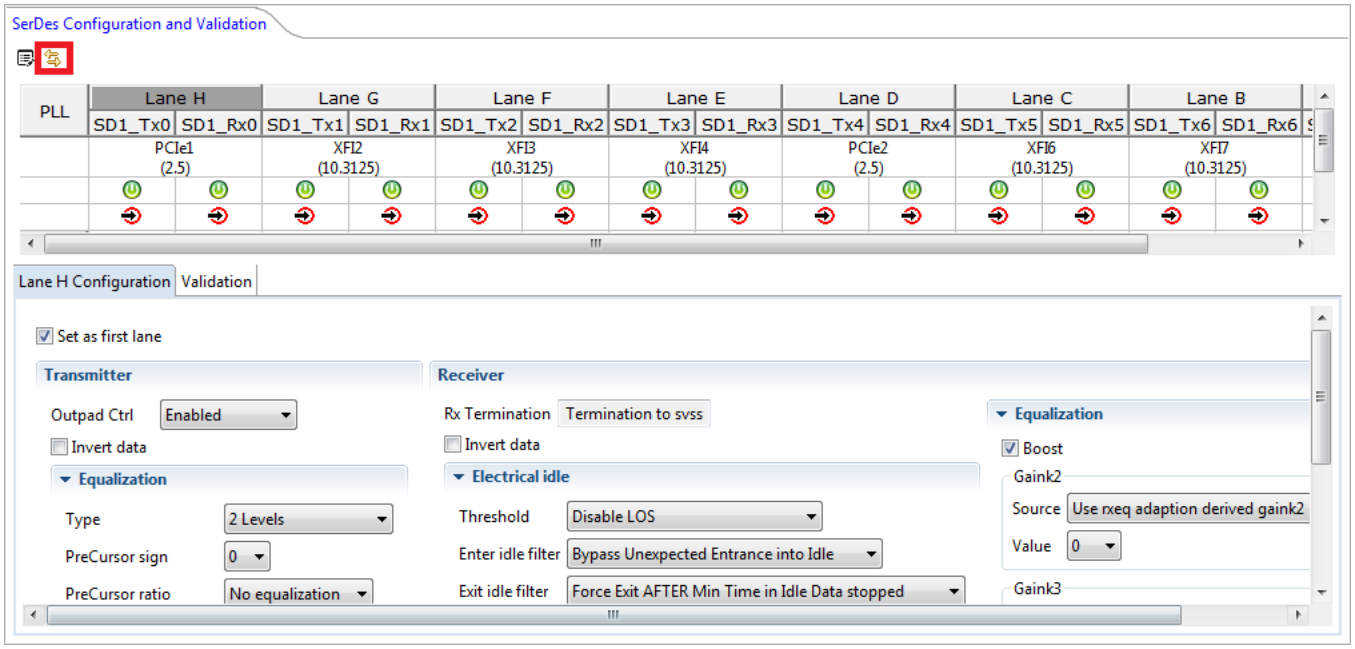
4.6. Synchronize PBL with other IP blocks

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:

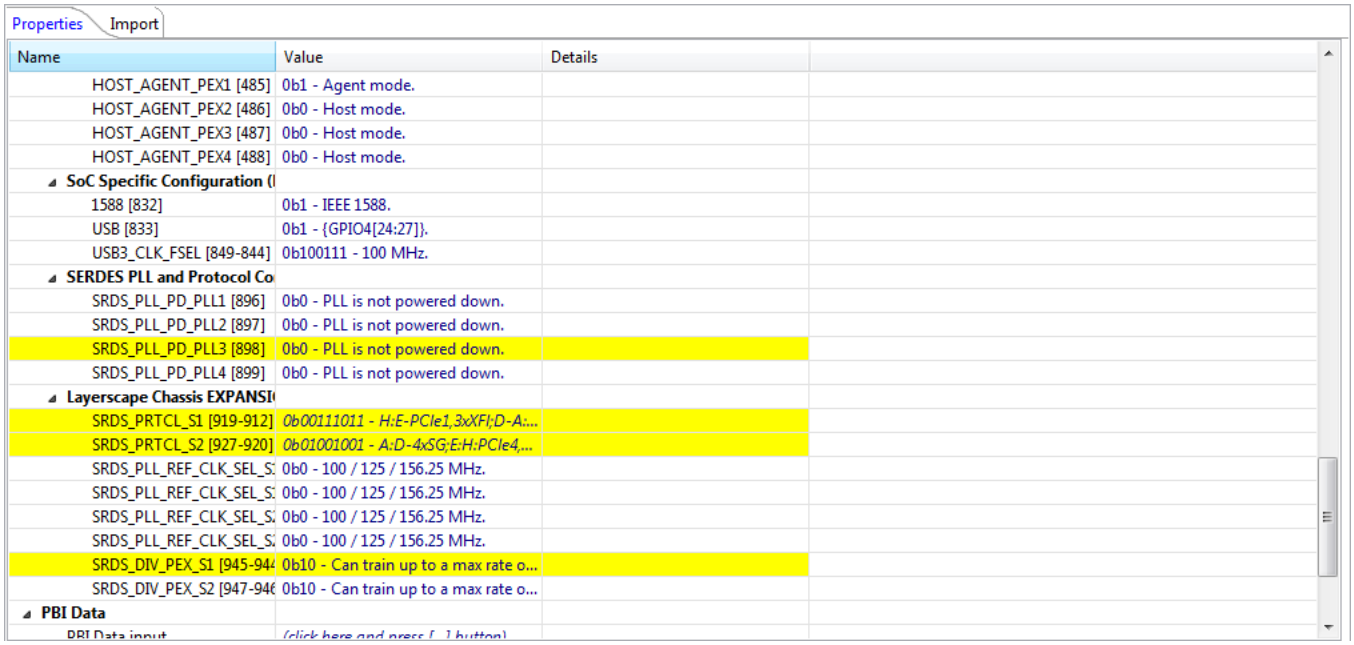
1. Ensure that a SerDes block component is available in the current project.
2. Double-click a SerDes component grouped under the SerDes block component in the **Components** view. The properties of the SerDes component are displayed on the **SerDes Configuration and Validation** page in the **Component 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.

Figure 20. Synchronizing PBL and SerDes components



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.

Figure 21. Verifying SerDes fields of PBL component

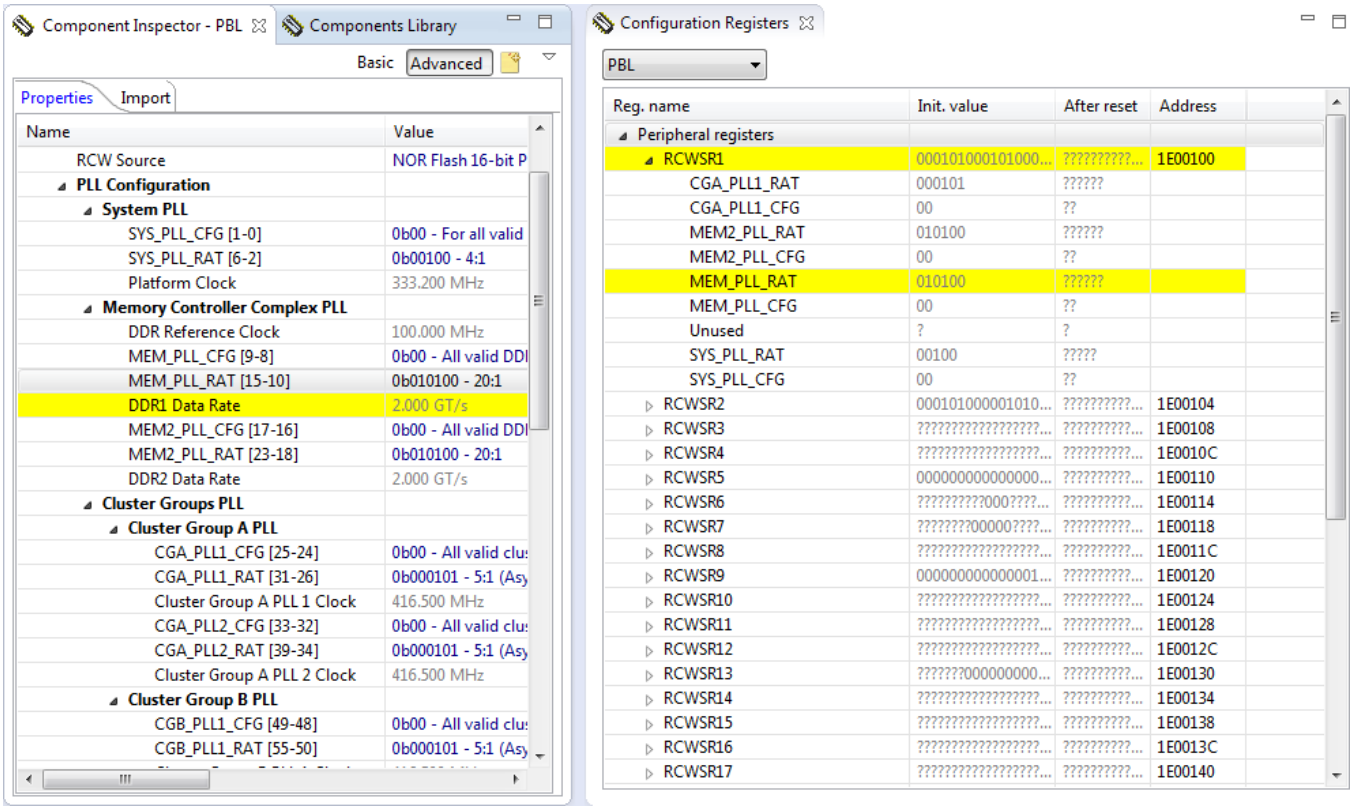


4.7. View RCW status registers

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

1. Choose **Window > Show View > Other** from the IDE menu bar. The **Show View** dialog appears.
2. Choose **Processor Expert > Configuration Registers** and click **OK**. The **Show View** dialog closes and the **Configuration Registers** view appears, displaying the details of the RCW status registers, as shown in the figure below.

Figure 22. Viewing RCW status registers



The **Configuration Registers** view reflects any changes made in the PBL configuration. Note that the don't care bits are displayed using the "?" character.

5. Advanced PBL operations

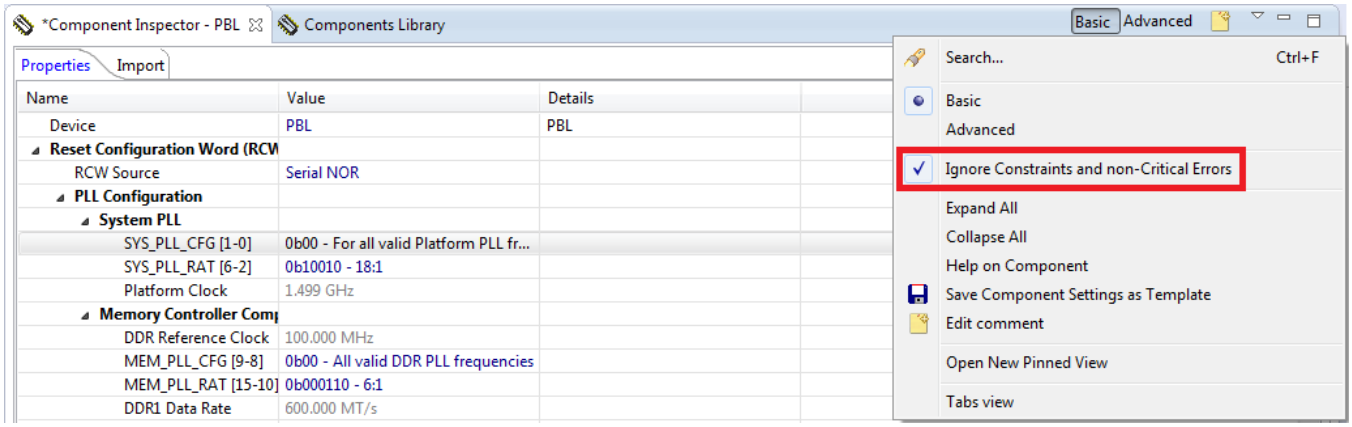
This section is divided into the following subsections:

5.1. Force RCW bit fields

You can make an RCW bit field have a value that is not supported in the PBL tool, by default. Follow these steps to force an RCW bit field to have unsupported value:

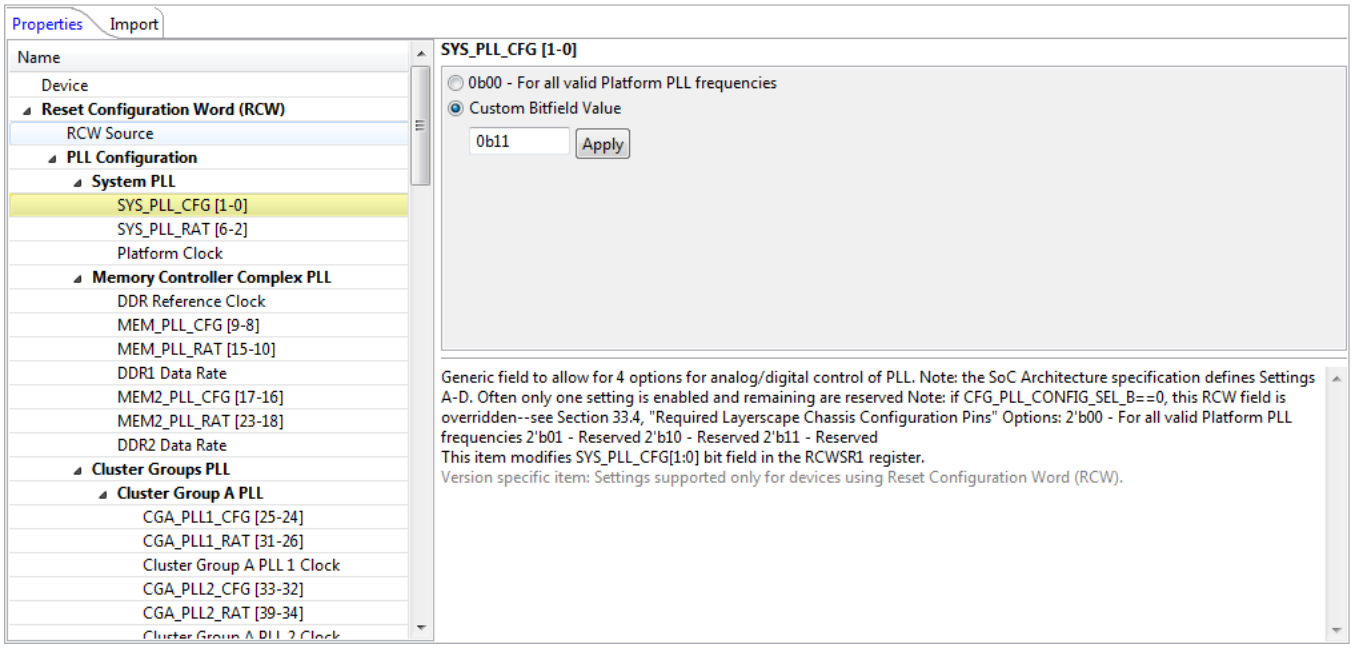
1. Click the **View Menu** button (down arrow button) on the toolbar of the **Component Inspector** view and choose **Ignore Constraints and non-Critical Errors**, as shown in the figure below.

Figure 23. Ignoring constraints and non-critical errors



2. Select an RCW bit field on the **Properties** page.
3. Select the **Custom Bitfield Value** option and specify a custom value for the RCW bit field, as shown in the figure below.

Figure 24. Specifying a custom RCW bit field value



4. Click **Apply**. The new value is added to the PBL configuration.

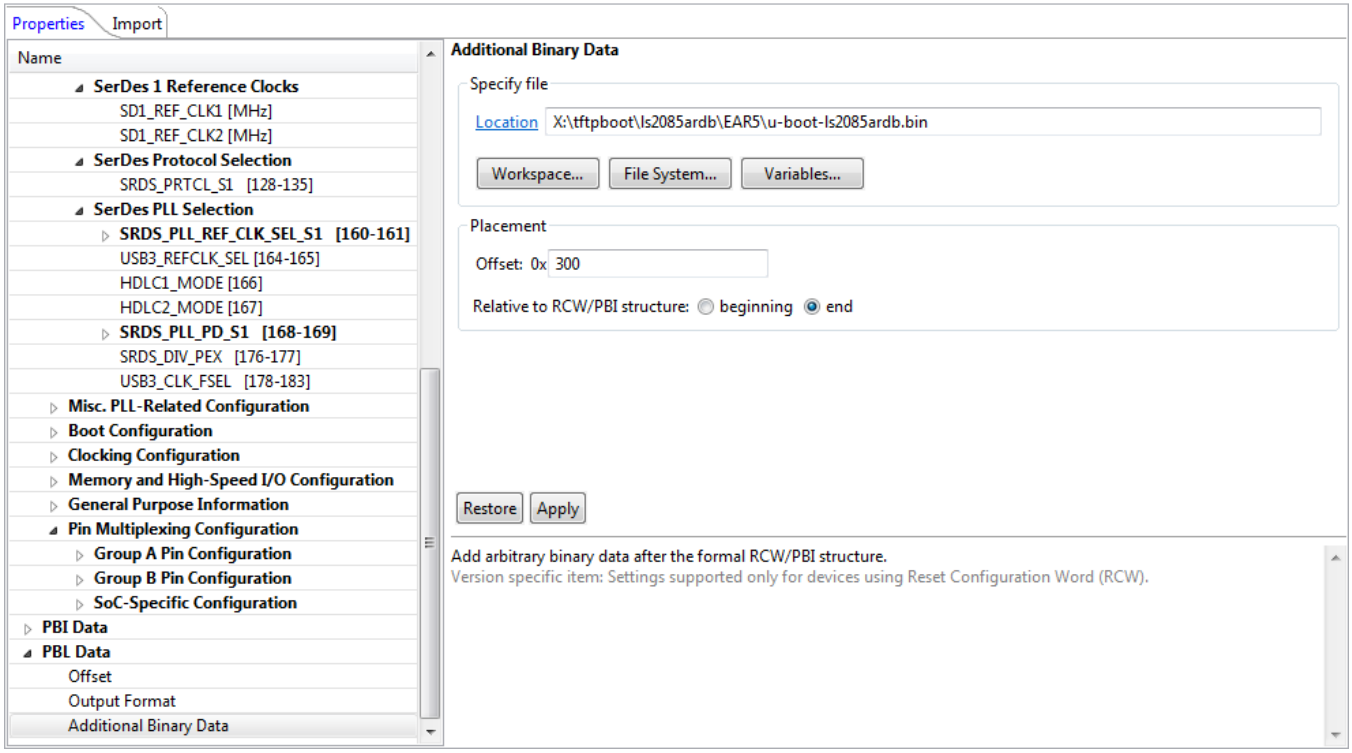
5.2. Add additional payload to a PBL image

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. Select the **Additional Binary Data** property under the **PBL Data** group in the **Name** column on the **Properties** page of the **Component Inspector** view.
2. Click the ellipsis (…) button in the **Value** column. The **Additional Binary Data** editor opens.
3. Specify a binary file to be added to the PBL image, in the **Location** field of the **Specify file** group.
4. Specify the offset and placement for the binary payload in the **Placement** group, as shown in the figure below.

Figure 25. Additional Binary Data editor



5. Click **Apply** to apply the changes.

5.3. Errata support

The PBL tool handles errata in the following two ways:

- If the PBL tool can implement the workaround for the erratum impacting the current PBL configuration, then the tool automatically implements the erratum workaround, without notifying the user. Examples of such errata include:
  - An RCW bit field option may not be available anymore
  - A chosen RCW bit field option may translate into a different value (as specified by the errata document of the chosen system-on-chip (SoC)) in the generated PBL binary

- If the PBL tool cannot implement the workaround for the erratum impacting the current PBL configuration, then the tool displays a warning message with the details of the erratum. An example is an erratum that depends on the conditions that cannot be determined by only assessing the PBL configuration (such as, operating temperature, dual inline package (DIP) switch configuration). For such an erratum, the user needs to manually implement the erratum workaround.

The warning message is displayed for the problematic field on the **Properties** page in the **Component Inspector** view and also in the **Problems** view, as shown in the following figures.

Figure 26. Warning displayed in Component Inspector view

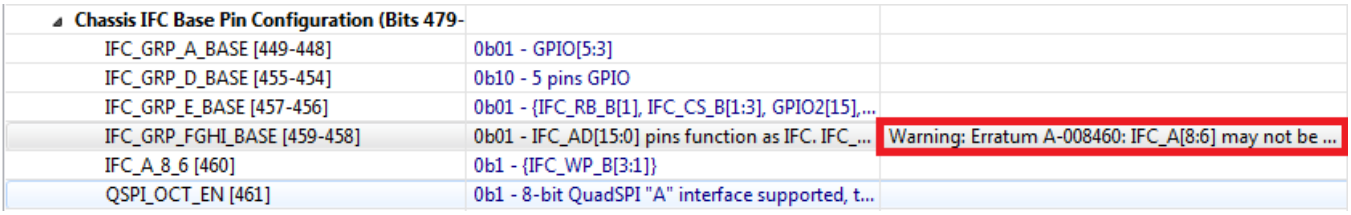
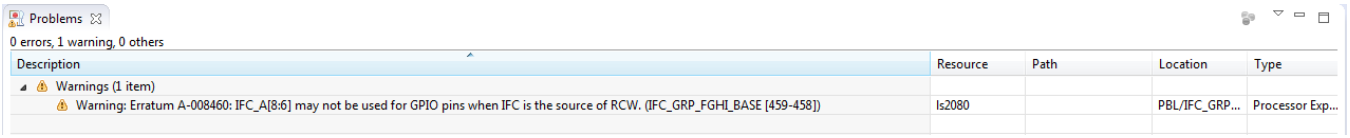


Figure 27. Warning displayed in Problems view



5.4. 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:

6.1. Reserved bits

When generating a PBL image, the PBL tool uses the default values of the bits that are marked as *Reserved* in the SoC reference manual. Due to this limitation, some mismatches may occur between the imported PBL image and generated PBL image.

6.2. 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.

How to Reach Us:

Home Page:
nxp.com

E-mail:
nxp.com/support

Document Number: AN5260

23 February 2016