Freescale Semiconductor
Application Note

Document Number: AN4948

Flash Programmer for CodeWarrior Power Architecture

1. Introduction

This application note describes the steps required to program a flash device using the CodeWarrior for Power Architecture.

This document includes the following section:

- Import or create a flash programmer target task
- Configure a flash programmer target task
- Add flash programmer actions on a NOR/NAND/SPI flash device
- Execute a flash programmer target task
- Program a flash device

Contents

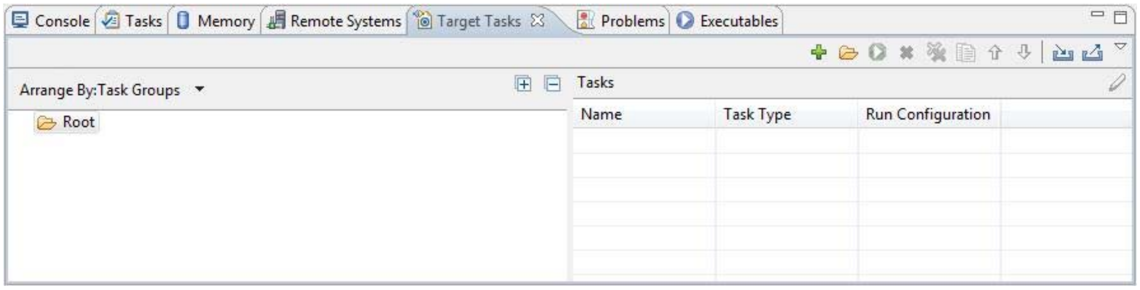
2. Preliminary background

The CodeWarrior flash programmer allows you to program flash memory device of the supported target boards from the CodeWarrior IDE. The flash programmer runs as a target task in the CodeWarrior IDE.

3. Configuring a flash device

To program a flash device using the CodeWarrior Flash Programmer, use the **Target Tasks** view. To open the **Target Tasks** view, choose **Window** > **Show View** > **Target Tasks** from the CodeWarrior IDE menu.

Figure 1. Target Tasks view



Now, you can use the **Target Tasks** view to:

- Import a target task or Create a target task
- Add flash programmer actions
- Execute flash programmer target task

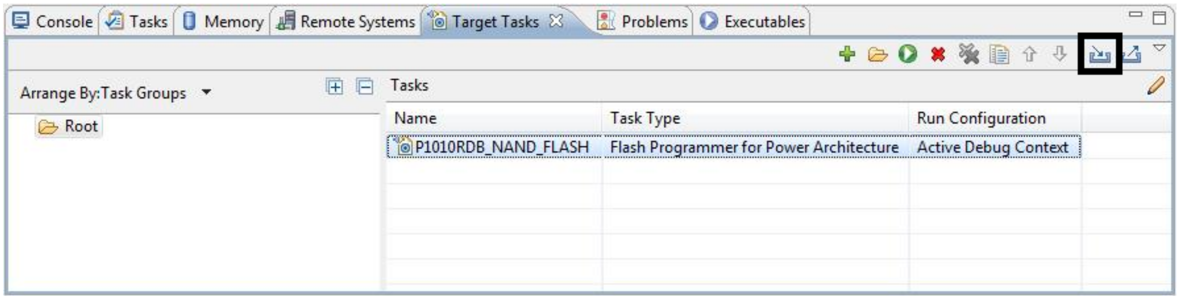
3.1. Import a target task

If you are using a Freescale board, you can use the flash programmer by importing an existing target task file, corresponding to the board type and flash device type (NAND, NOR, or SPI).

To import a target task file, perform the following steps:

1. From **Target Tasks** view toolbar, choose **Import** ( ) and browse to `<CWInstallDir>\PA\bin\plugins\support\TargetTask\Flash_Programmer`
2. Select the target task file corresponding to the board type and flash device type.

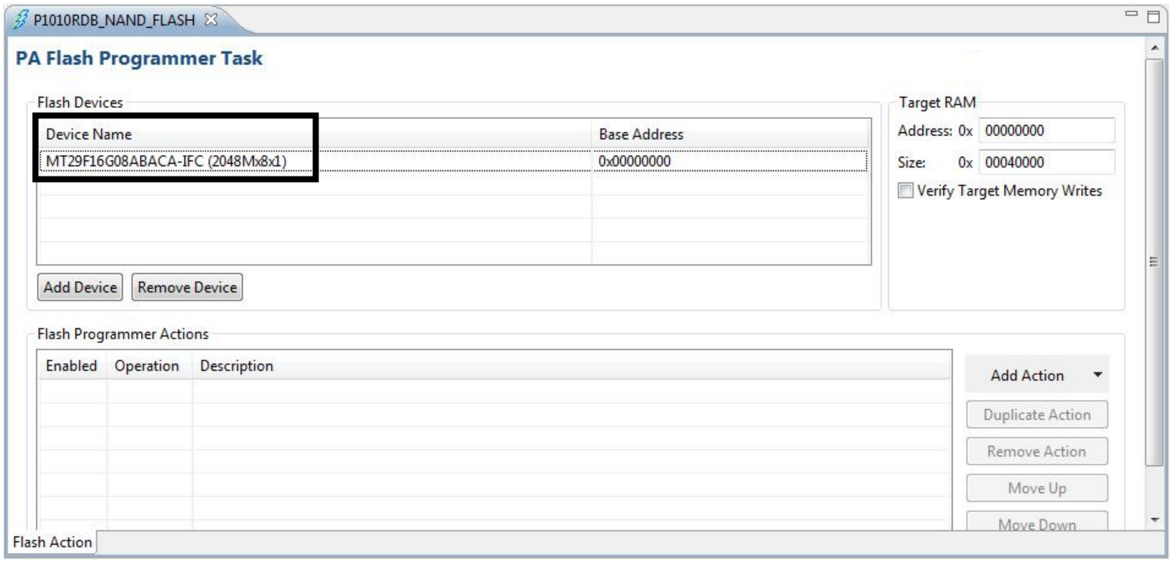
Figure 2. Importing a target task



Name	Task Type	Run Configuration
P1010RDB_NAND_FLASH	Flash Programmer for Power Architecture	Active Debug Context

3. Double-click the imported target task in the **Target Tasks** view to display the target task details in a new window (see Figure 3).

Figure 3. Target task details



Device Name	Base Address
MT29F16G08ABACA-IFC (2048Mx8x1)	0x00000000

Address: 0x 00000000
Size: 0x 00040000
Verify Target Memory Writes

Enabled	Operation	Description

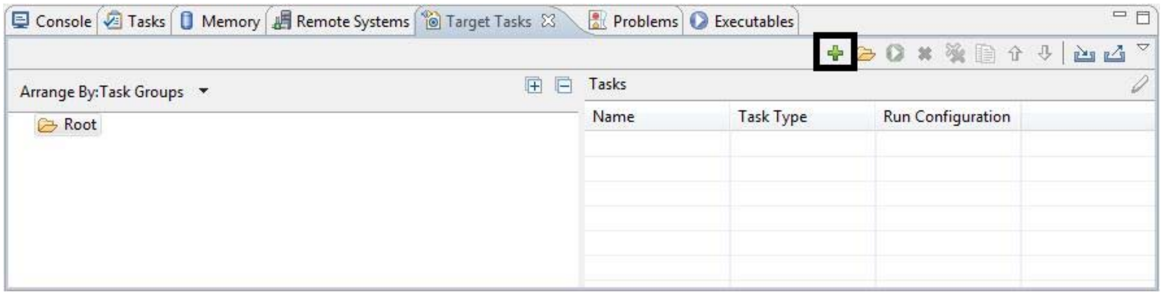
In the figure above, you can notice that a flash device is added to the imported target task. You can add flash programmer actions to the flash device. See Add flash programmer actions to add flash programmer actions.

3.2. Create a target task

When a custom board is used, it is recommended to create a new target task.

1. In the **Target Tasks** view, click **Create New Target Task** (✚), as shown in Figure 4. The **New Target Task** dialog appears (see Figure 5).

Figure 4. Creating a target task

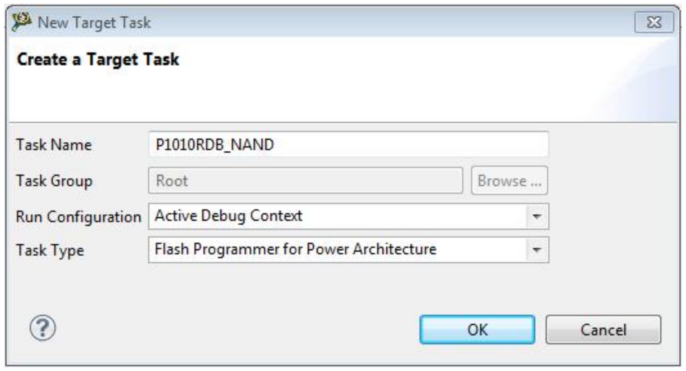


Name	Task Type	Run Configuration

2. In the **New Target Task** dialog, enter a target task name in the **Task Name** text box, choose a launch configuration from the **Run Configuration** menu, and choose **Flash Programmer for Power Architecture** from the **Task Type** menu.

   - Choose **Active Debug Context** as your launch configuration when flash programmer is used over an active debug session.
   - Choose a project-specific debug context as your launch configuration when flash programmer is used without an active debug session.

Figure 5. New Target Task dialog

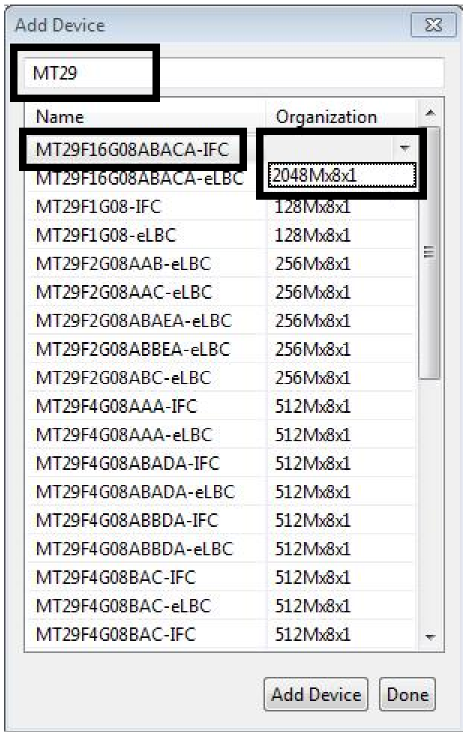

New Target Task

Create a Target Task

Task Name	P1010RDB_NAND
Task Group	Root
Run Configuration	Active Debug Context
Task Type	Flash Programmer for Power Architecture

3. Click **OK**. The **New Target Task** dialog closes and a new target task is created in the **Target Tasks** view.
4. Double-click the newly created target task in the **Target Tasks** view. A new window similar to one shown in Figure 3 displays, where you can configure the new target task.
5. Click **Add Device** to add a flash device to the target task. The **Add Device** dialog appears (see Figure 6).
6. Specify the name and organization of the flash device you want to add to the target task and click **Add Device**, as shown in Figure 6.

Figure 6. Add Device dialog


Add Device

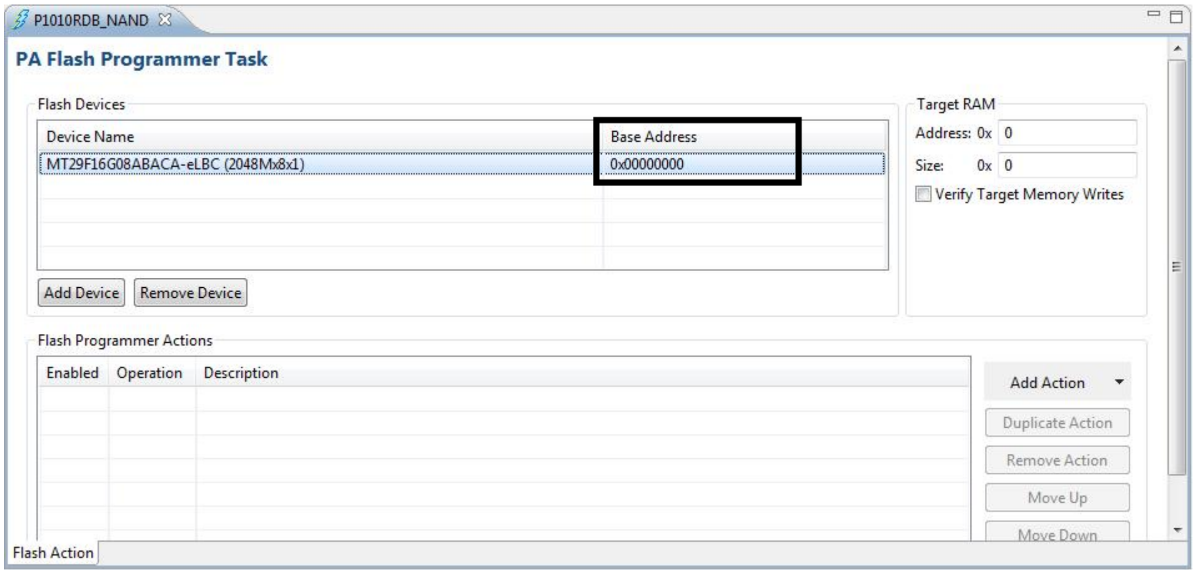
MT29

Name	Organization
MT29F16G08ABACA-IFC	
MT29F16G08ABACA-eLBC	2048Mx8x1
MT29F1G08-IFC	128Mx8x1
MT29F1G08-eLBC	128Mx8x1
MT29F2G08AAB-eLBC	256Mx8x1
MT29F2G08AAC-eLBC	256Mx8x1
MT29F2G08ABAEA-eLBC	256Mx8x1
MT29F2G08ABBEA-eLBC	256Mx8x1
MT29F2G08ABC-eLBC	256Mx8x1
MT29F4G08AAA-IFC	512Mx8x1
MT29F4G08AAA-eLBC	512Mx8x1
MT29F4G08ABADA-IFC	512Mx8x1
MT29F4G08ABADA-eLBC	512Mx8x1
MT29F4G08ABBDA-IFC	512Mx8x1
MT29F4G08ABBDA-eLBC	512Mx8x1
MT29F4G08BAC-IFC	512Mx8x1
MT29F4G08BAC-eLBC	512Mx8x1
MT29F4G08BAC-IFC	512Mx8x1

7. Click **Done**. The **Add Device** dialog closes and the specified flash device is added in the target task details window (see Figure 7).

8. Specify a base address for the flash device, as shown in Figure 7.

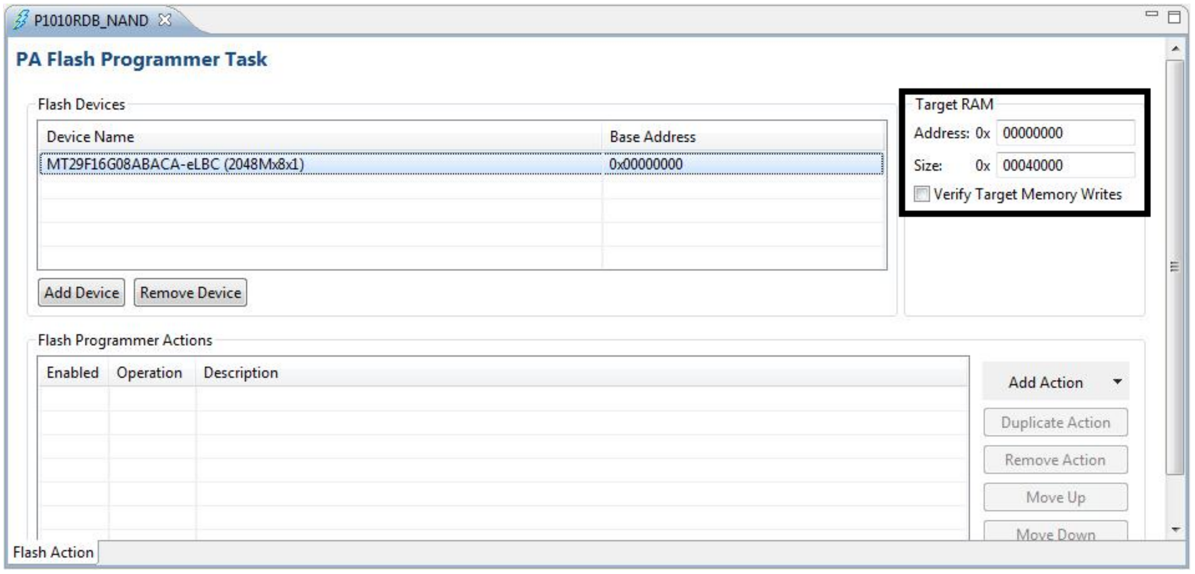
Figure 7. Specifying base address for a flash device



NOTE	For NOR flashes, base address indicates a location where the flash is mapped in the memory. For NAND and SPI flashes, the base address is usually `0x0`.

9. Specify target RAM settings for the flash device, as shown in Figure 8. The target RAM is used by the flash programmer to download its algorithm.

Figure 8. Specifying target RAM for a flash device



NOTE The target RAM memory area is not restored by the flash programmer. If the flash programmer is used with an active debug context, it impacts the debug session.

3.3. Add flash programmer actions

Following actions can be added on a flash device:

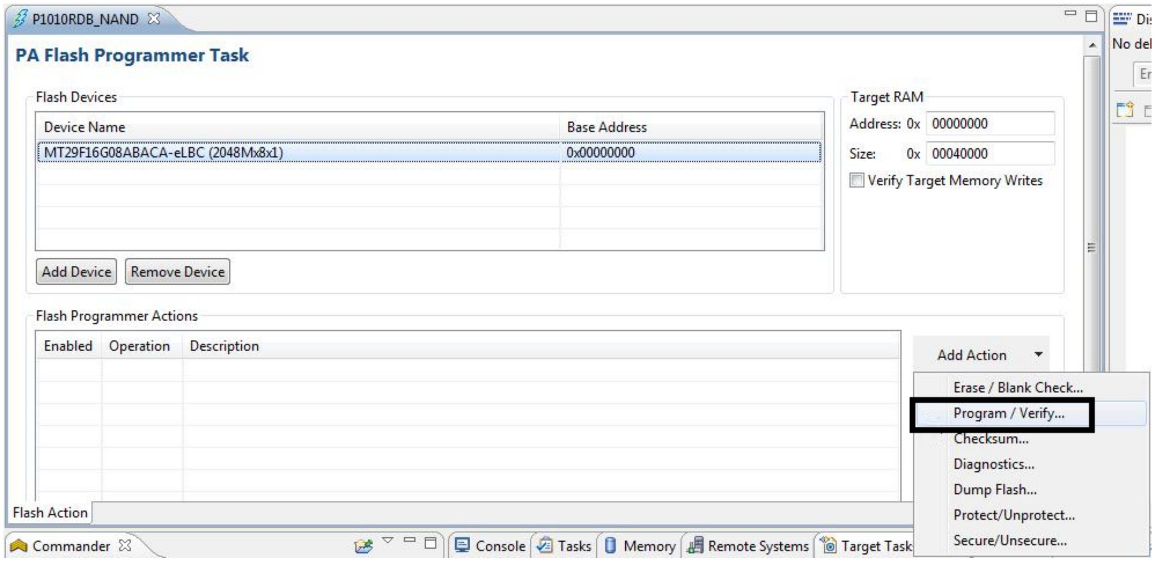
- Erase/blank check
- Program/verify
- Checksum
- Diagnostics
- Dump flash
- Protect/unprotect
- Secure/unsecure

For details on each of these actions, see *CodeWarrior Development Studio for Power Architecture® Processors Version 10.x Tracing and Analysis Tools User Guide*, available in the `<CWInstallDir>\PA\Help\PDF` folder.

Follow these steps to add a flash programmer action on a flash device:

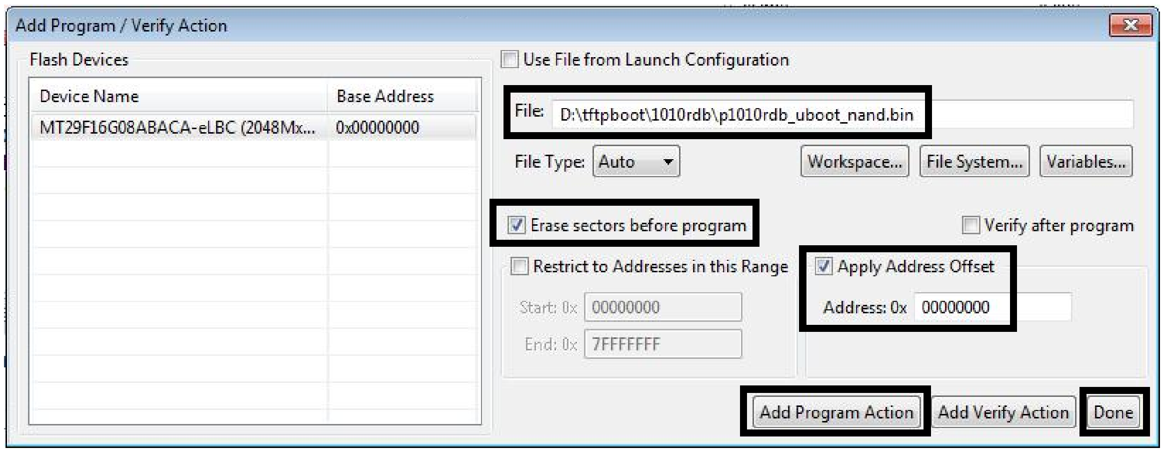
1. Choose **Program / Verify** from the **Add Action** menu in the target task details window, as shown in Figure 9. The **Add Program / Verify Action** dialog appears (see Figure 10).

Figure 9. Adding a flash programmer action on a flash device



2. Select the complete path and type of the file, to be written to the flash device. If a separate erase action was not used before using the program/verify action, then select the **Erase sectors before program** option. Also, ensure that the correct address offset is specified, so that the image is programmed into the correct location.

Figure 10. Add Program / Verify Action dialog



Device Name	Base Address
MT29F16G08ABACA-eLBC (2048Mx...	0x00000000

File: D:\tftpboot\1010rdb\p1010rdb_uboot_nand.bin

File Type: Auto

[✓] Erase sectors before program

[ ] Verify after program

[ ] Restrict to Addresses in this Range

Start: 0x 00000000

End: 0x 7FFFFFFF

[✓] Apply Address Offset

Address: 0x 00000000

NOTE The **Verify after program** option is available, but only if the processor supports it.

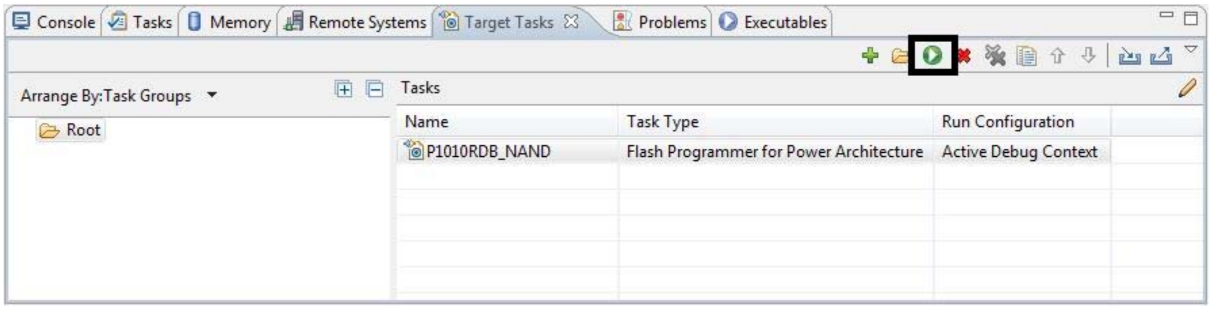
3. Select **Add Program Action**, to add the action, and then select **Done** to close the **Add Program / Verify Action** dialog.

3.4. Execute flash programmer target task

Once you have finished configuring your flash programmer target task, you can execute the target task. To execute a flash programmer target task, perform the following steps:

1. Select the target task in the **Target Tasks** view and click **Execute ( )**, as shown in Figure 11.

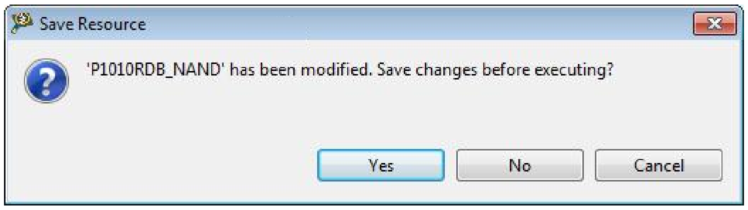
Figure 11. Executing target task



Name	Task Type	Run Configuration
P1010RDB_NAND	Flash Programmer for Power Architecture	Active Debug Context

2. If prompted, select **Yes** in the **Save Resource** dialog, to ensure that changes are saved before executing the target task.

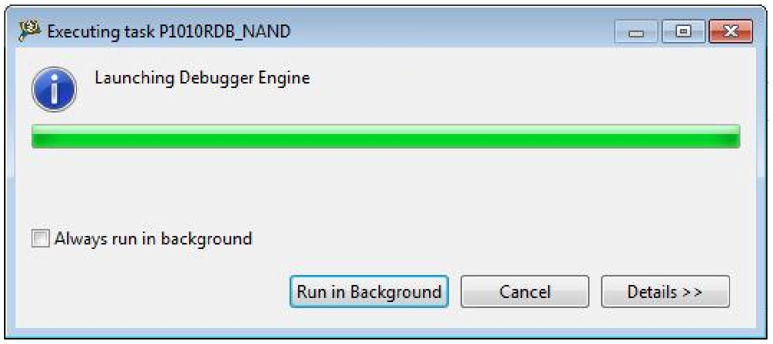
Figure 12. Save Resource dialog



'P1010RDB_NAND' has been modified. Save changes before executing?

The flash programmer task starts executing, as shown in Figure 13.

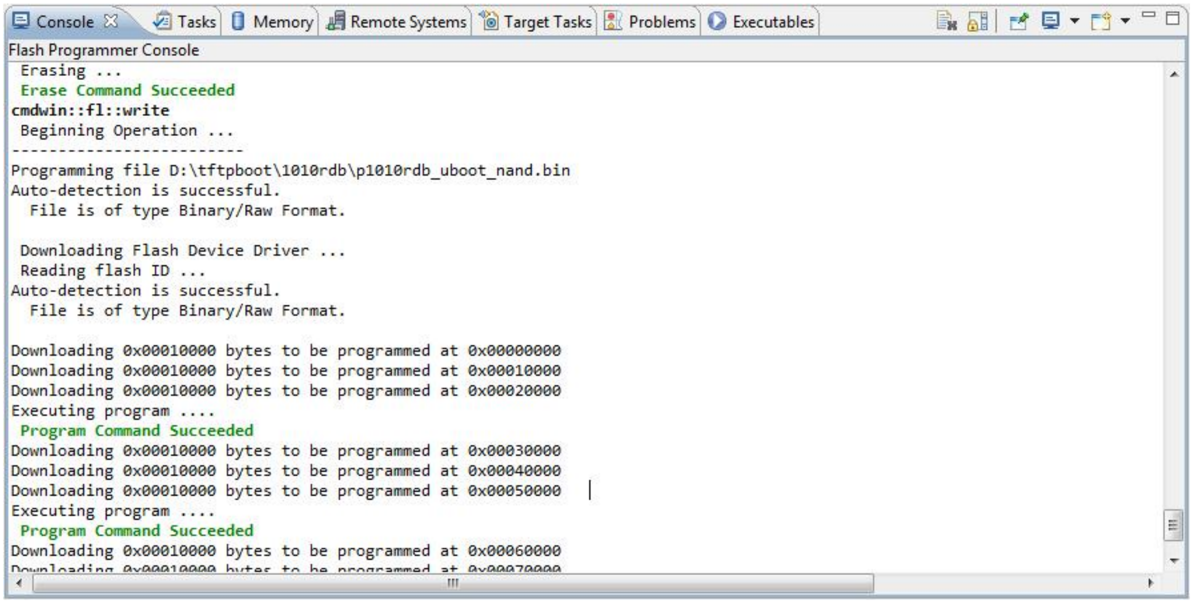
Figure 13. Executing task dialog



Launching Debugger Engine

You can check the results of flash actions in the **Console** view (see Figure 14). The green color indicates success of the task and the red color indicates failure of the task.

Figure 14. Console view



```
 Erasing ...
 Erase Command Succeeded
cmdwin::fl::write
 Beginning Operation ...
-------------------------
Programming file D:\tftpboot\1010rdb\p1010rdb_uboot_nand.bin
Auto-detection is successful.
  File is of type Binary/Raw Format.

 Downloading Flash Device Driver ...
 Reading flash ID ...
Auto-detection is successful.
  File is of type Binary/Raw Format.

Downloading 0x00010000 bytes to be programmed at 0x00000000
Downloading 0x00010000 bytes to be programmed at 0x00010000
Downloading 0x00010000 bytes to be programmed at 0x00020000
Executing program ....
 Program Command Succeeded
Downloading 0x00010000 bytes to be programmed at 0x00030000
Downloading 0x00010000 bytes to be programmed at 0x00040000
Downloading 0x00010000 bytes to be programmed at 0x00050000
Executing program ....
 Program Command Succeeded
Downloading 0x00010000 bytes to be programmed at 0x00060000
Downloading 0x00010000 bytes to be programmed at 0x00070000
```

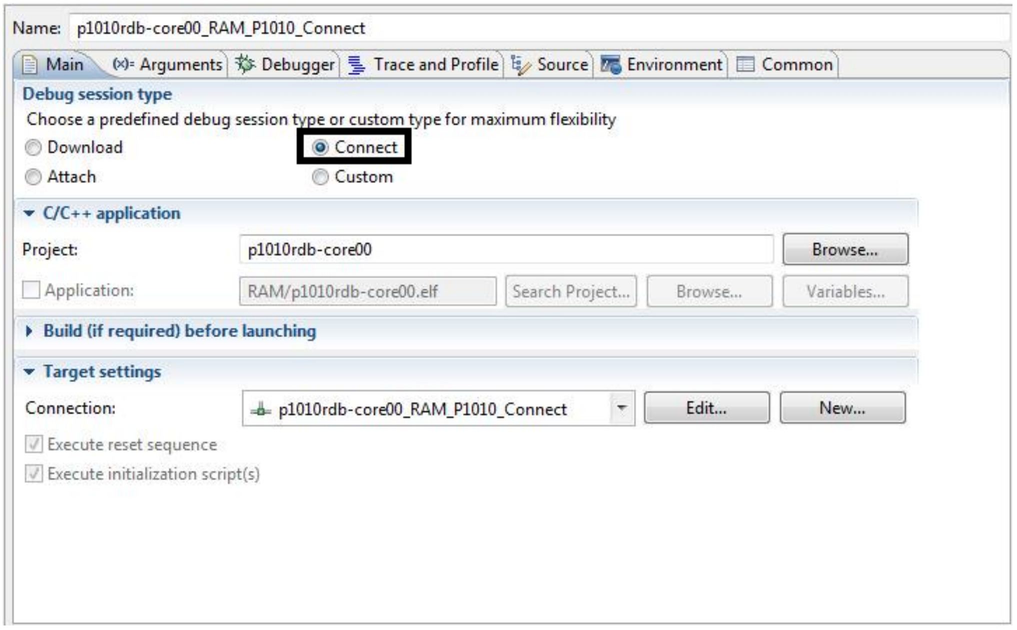
4. Programming a flash device

During bring-up and board support procedures, programming using flash programmer usually fails. The reason is that the DDR memory is not initialized or configured incorrectly. Because each new processor has an internal memory, you can use an SRAM initialization file with **Connect** a target and flash programmer, to program a flash devices.

To program a flash device, perform the following steps:

1. Choose **Run** > **Debug Configurations** from the CodeWarrior IDE menu bar. The **Debug Configurations** dialog appears.
2. Create a new debug configuration or edit an existing configuration, select **Connect** as your debug session type, as shown in Figure 15.

Figure 15. Selecting debug session type in Debug Configurations dialog



Name: p1010rdb-core00_RAM_P1010_Connect

Debug session type

Choose a predefined debug session type or custom type for maximum flexibility

Download [✓ Connect] Attach Custom

C/C++ application

Project: p1010rdb-core00

Application: RAM/p1010rdb-core00.elf

Build (if required) before launching

Target settings

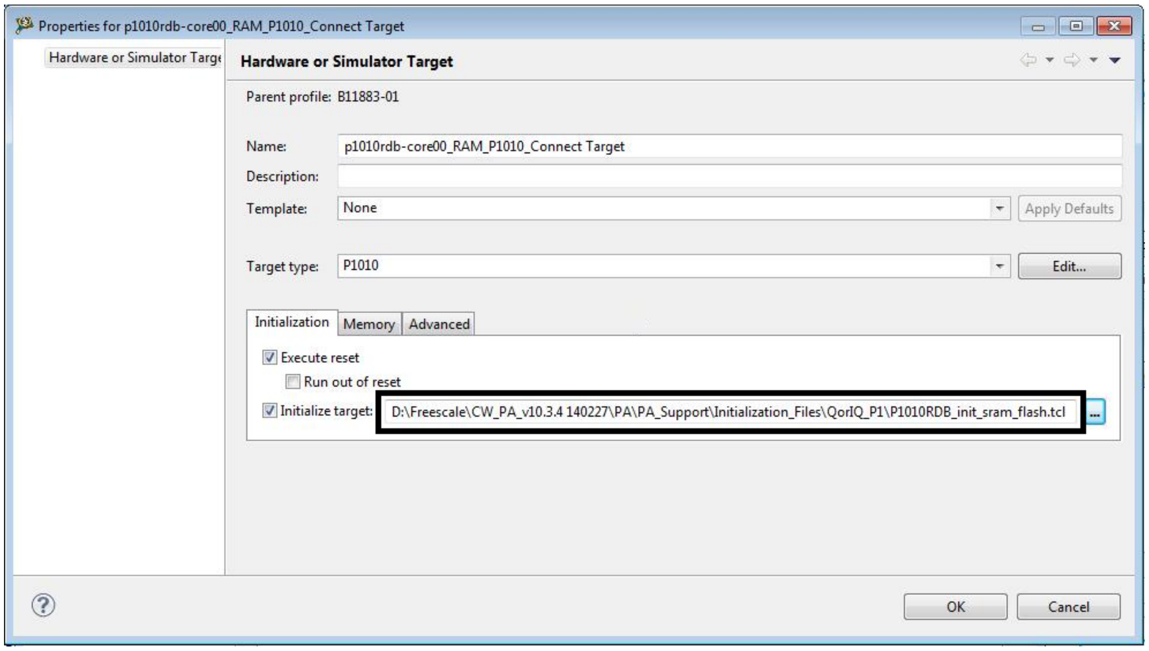
Connection: p1010rdb-core00_RAM_P1010_Connect

✓ Execute reset sequence

✓ Execute initialization script(s)

3. Click **Edit** next to the **Connection** menu. The **Properties** window appears (see Figure 16).
4. Browse to the SRAM initialization file located inside the `<CWInstallDir>\PA\PA_Support\Initialization_Files` folder, as shown in Figure 16.

Figure 16. Properties window



Properties for p1010rdb-core00_RAM_P1010_Connect Target

Hardware or Simulator Target

Parent profile: B11883-01

Name: p1010rdb-core00_RAM_P1010_Connect Target

Description:

Template: None

Target type: P1010

Initialization | Memory | Advanced

☑ Execute reset

☐ Run out of reset

☑ Initialize target: D:\Freescale\CW_PA_v10.3.4 140227\PA\PA_Support\Initialization_Files\QorIQ_P1\P1010RDB_init_sram_flash.tcl

5. Open a debug session using the above settings and use the flash programmer to program the flash device.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

Document Number: AN4948

5 May 2014