i.MX31 PDK 1.4 Windows CE 5.0

User's Guide

Document Number: 926-77197
Rev. 1.4
10/2008

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 010 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

About This Book

This guide explains how to build and execute Windows® CE™ 5.0 OS images for the 3-Stack board, using the Freescale 3-Stack Windows CE 5.0 Board Support Package (BSP).

Audience

This document is intended for software, hardware, and system engineers who are planning to use the product and for anyone who wants to understand more about the product.

Organization

This document contains the following chapters.

Chapter 1 "BSP Installation" explains how to install/uninstall the BSP.

Chapter 2 "BSP Contents and Organization" describes the contents and organization of the BSP.

Chapter 3 "Configuring OS Images" explains how to configure the BSP for building Windows CE OS images.

Chapter 4 "Building OS Images" explains how to build Windows CE OS images using the BSP.

Chapter 5 "Preparing for Downloading and Debugging" describes how to prepare for downloading and debugging.

Chapter 6 "Windows CE Applications in the BSP" identifies the Windows CE 5.0 projects you can create with the BSP.

Conventions

This document uses the following conventions:

`Courier`	Is used to identify commands, explicit command parameters, code examples, expressions, data types, and directives.
Italic	Is used for emphasis, to identify new terms, and for replaceable command parameters.

References

The following documents were referenced to build this document.

1. i.MX31 3-Stack Platform System User's Guide

2. i.MX Platform Advanced ToolKit Standard Version User's Guide

Chapter 1
BSP Installation

The Freescale 3-Stack Board Support Package (BSP) is a collection of binary, code, and support files that you use to create Kernel images for the 3-Stack board. The BSP is distributed as a single archive MSI (.msi) file.

1.1 Installing the BSP

This section explains how to install Platform Builder, and then install the BSP into the Windows CE source code tree and the Platform Builder development environment.

1.1.1 Install Platform Builder

To install Platform Builder, use these steps:

1. Insert the Windows CE 5.0 installation disk, or download Platform Builder from the following link http://www.microsoft.com/downloads/details.aspx?familyid=486E8250-D311-4F67-9FB3-23E8B8944F3E&displaylang=en.
2. From the Start menu of the installation disk, select the Install option (Figure 1-1).

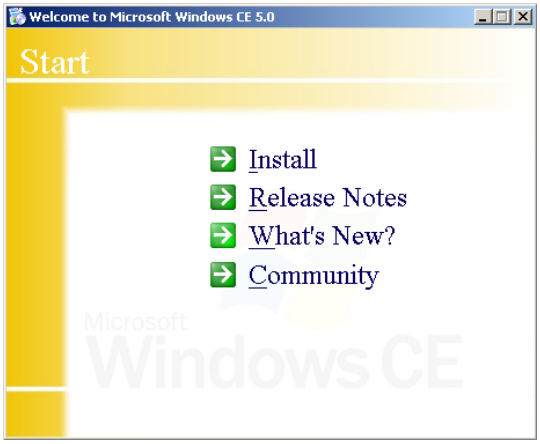

Figure 1-1 Start Menu

3. Follow the installation wizard until you see the “Setup Type” window (Figure 1-2).

4. Select **Custom** (**Tools and OS**), and then click **Next**.
5. Select the destination driver for the Platform Builder installation.

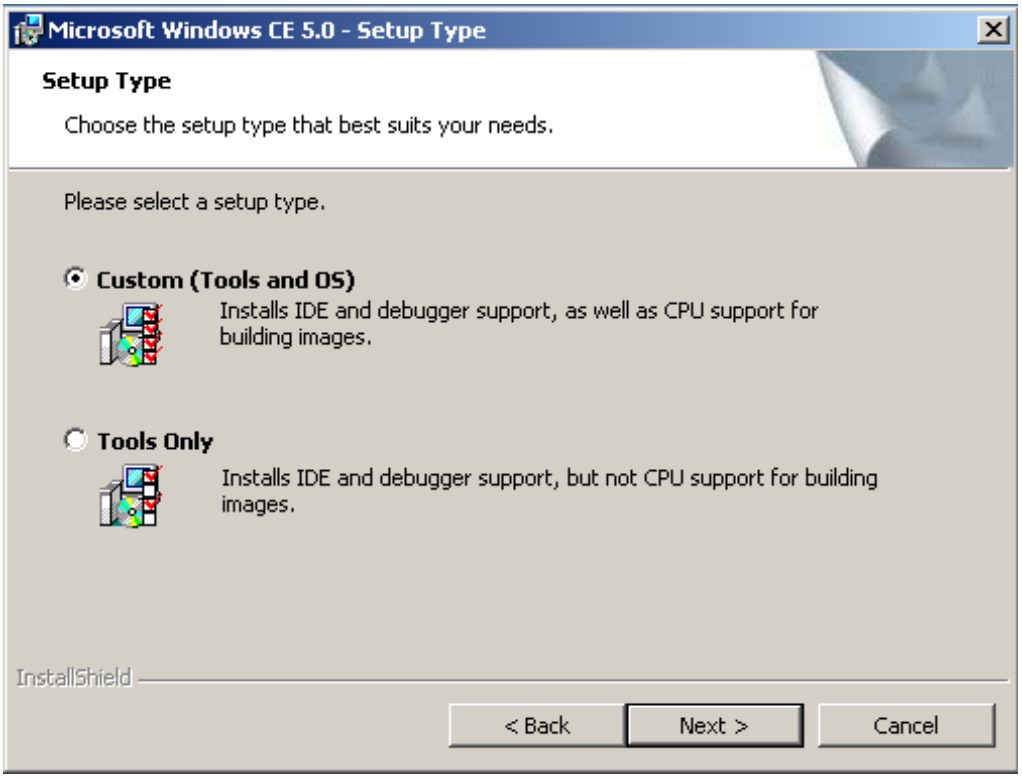

Figure 1-2 Setup Type Window

6. At the Custom Setup window, select the options shown in Figure 1-3.

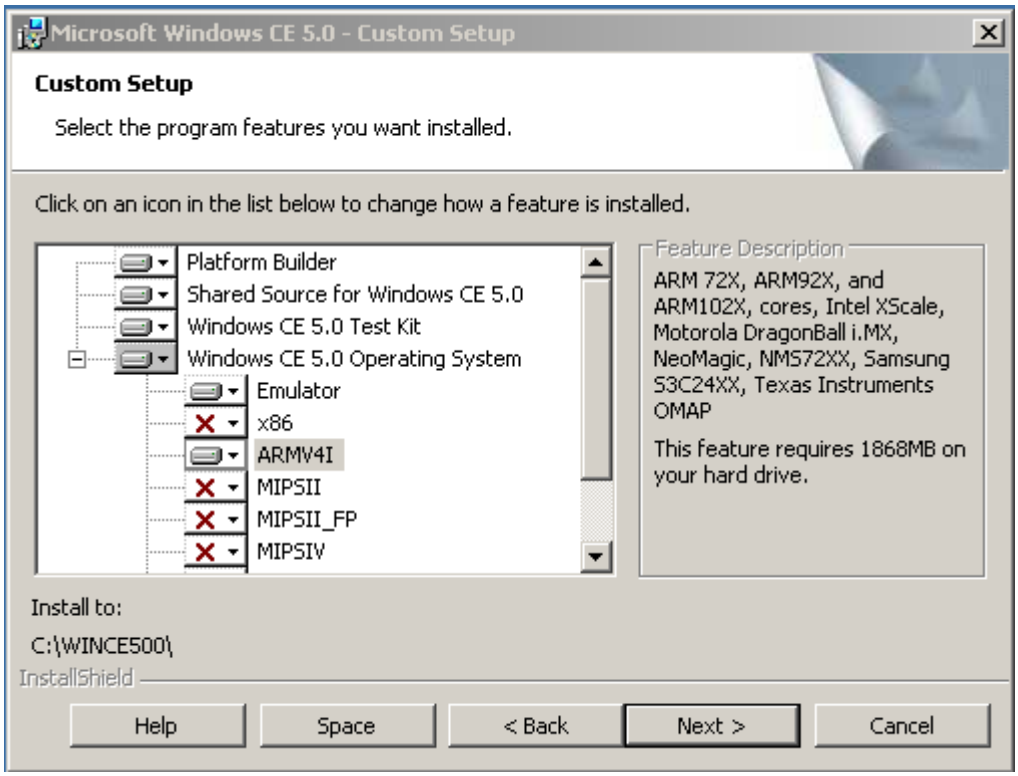

Figure 1-3 Custom Setup Window

Note

Be sure to select the ARM41™ compiler option from the Windows CE 5.0 Operating System folder. This is very important, because the i.MX31 Windows CE Software Development Kit (SDK) uses this to compile the BSP files.

7. Follow the installation wizard until the installation is complete.
8. Install the updates in the list that follows. These are required by the SDK.

Windows CE 5.0 Platform Builder Cumulative Product Update Rollup Package (through 12/31/07)

http://www.microsoft.com/downloads/details.aspx?FamilyID=a54779d5-f4a5-49f0-9e36-979d461f536c&DisplayLang=en

Windows CE 5.0 Platform Builder Monthly Update (January 2008)

http://www.microsoft.com/downloads/details.aspx?FamilyId=98F0906A-45C6-4792-91FE-47A0E073A998&displaylang=en

Windows CE 5.0 Platform Builder Monthly Update (February 2008)

http://www.microsoft.com/downloads/details.aspx?FamilyID=9ac7441f-3adc-4e35-abd1-fa4fbb90a132&DisplayLang=en

Windows CE 5.0 Platform Builder Monthly Update (March 2008)

http://www.microsoft.com/downloads/details.aspx?FamilyID=aacea4ae-e8ce-40c3-941c-d2a549a55deb&DisplayLang=en

Windows CE 5.0 Platform Builder Monthly Update (April 2008)

http://www.microsoft.com/downloads/details.aspx?FamilyID=4a3d4537-ab1b-4e8f-b735-348a64a9d8d6&DisplayLang=en

1.1.2 Building the SD Card Drivers from Microsoft

The updates for Platform Builder modify the SD Card drivers from Microsoft, and you must recompile the SD Card drivers for the i.MX31 Windows CE SDK to work. This step is a one-time process; there is no need to repeat this each time you build a new OS image.

To recompile the drivers, use these steps:

1. Open a Platform Builder 5.0 session.
2. Open a Platform Builder's Workspace for the i.MX31 3-Stack board. The BSP provides a sample workspace that can be used for this example.
3. On the File View tab, right-click on:

   `\WINCE500\PUBLIC\COMMON\OAK\DRIVERS\SDCARD`

4. Select **Clean Before Building**, and then select **Build Current Project** (see both in Figure 1-4).

   The Platform Builder build process is complete.

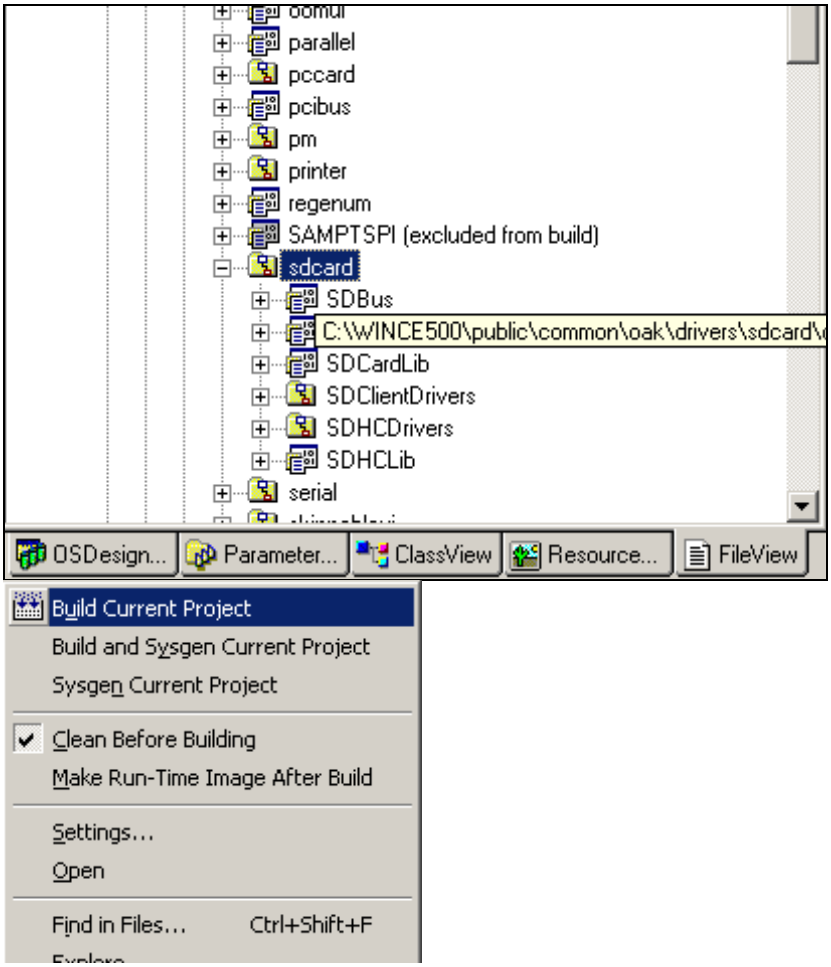

Figure 1-4 Clean Before Building and Build Current Project

1.2 Removing an Existing Installed BSP

This section explains how to remove an installation of the BSP from the Windows CE source code tree and Platform Builder development environment.

NOTE

Before uninstalling the BSP, save any modified files that you want to keep to a protected location, because uninstalling the BSP will remove all files that were populated by the Microsoft Windows Installer.

To remove the BSP, use these steps:

1. Close Platform Builder.

2. Right-click on the Microsoft Windows Installer (.msi) file and select **Uninstall**.

3. Manually remove the remaining BSP files and directories that were not uninstalled by the previous step:

```
\WINCE500\PBWorkspaces\3DSmobility
\WINCE500\PLATFORM\COMMON\SRC\ARM\dirs
\WINCE500\PLATFORM\COMMON\SRC\ARM\FREESCALE
\WINCE500\PLATFORM\3DS
\WINCE500\PUBLIC\COMMON\OAK\CATALOG\CEC\mx313ds.cec
\WINCE500\PUBLIC\COMMON\OAK\CSP\ARM\dirs
\WINCE500\PUBLIC\COMMON\OAK\CSP\ARM\FREESCALE
\WINCE500\SUPPORT
```

4. Manually remove the residual BSP library files.

   To find the libraries, use Windows Explorer with the search name *mxarm11*;*mx31*;*pmic*;*mc13783* for the following two library paths and remove all of the LIB, PDB, and DEF files found by the search:

```
\WINCE500\PUBLIC\COMMON\OAK\LIB\ARMV4I
\WINCE500\PLATFORM\COMMON\LIB\ARMV4I
```

5. Remove the BSP catalog.
   - In Platform Builder, click **File** > **Manage Catalog Items**. The Manage Catalog items dialog is displayed.
   - Find the **mx313ds.cec** catalog item, select the entry and click **Remove**.

1.3 Importing the BSP Catalog and Sample Workspace

After installing the BSP, you can use the sample workspace provided in the BSP package to build a Windows CE OS image based on the BSP.

To build a Windows CE OS image based on the BSP, use these steps:

1. In Platform Builder, click **File > Manage Catalog Items**.

   The Manage Catalog items dialog is displayed.

2. Click **Import**, and then select the following catalog file:

```
WINCE500\PUBLIC\COMMON\OAK\CATALOG\CEC\mx313ds.cec
```

3. Click **Open**, and then click **OK** to import the catalog.

4. In Platform Builder, click **File > Open Workspace**.

5. Select the BSP sample workspace:

```
\WINCE500\PBWorkspaces3DSmobility\3DSmobility.pbxml
```

   To view the loaded workspace, click **View > Workspace**.

Chapter 2
BSP Contents and Organization

The BSP is a collection of code and support files that can be integrated into the Microsoft Platform Builder development environment to create Windows CE OS images for the 3-Stack board.

The BSP contains the following elements:

- Boot loader for downloading OS images
- OEM Adaptation Layer (OAL) for providing the kernel hardware interface
- Device drivers to support on-chip and on-board peripherals
- Image configuration and build files

The Chip Support Package (CSP) and Production Quality OAL (PQOAL) are shared by all i.MX31-based platforms.

2.1 About the Chip Support Package (CSP)

The Freescale CSP directory contains a collection of chipset-level code that you can leverage to develop platforms based on the i.MX31 system on a chip (SoC). The driver code and definitions in the CSP can be reused in a new platform design for the i.MX31 without modification. To keep the CSP platform-agnostic, drivers in the CSP directory utilize hardware abstraction routines that can be ported to a specific platform or board. The CSP source code is compiled into a set of static libraries that are ultimately linked with platform-specific libraries to create drivers for the system.

The following directories contain the CSP source code:

```
WINCE500\PUBLIC\COMMON\OAK\CSP\ARM\FREESCALE\MXARM11
WINCE500\PUBLIC\COMMON\OAK\CSP\ARM\FREESCALE\MX31
```

The driver code is reusable:

- The CSP driver code in the `MXARM11` directory is reusable across all Freescale ARM11™-based SoCs.
- The CSP driver code in the `MX31` directory is reusable across all platforms based on the MX31 device.

2.2 About the Production Quality OAL (PQOAL)

Where possible, the BSP leverages the PQOAL (production-quality OEM adaptation layer) architecture and components provided by Microsoft to reduce the OAL code that needs to be modified and maintained by the Original Equipment Manufacturer (OEM). In addition, PQOAL components customized for the i.MX31 are available in the following directories:

```
WINCE500\PLATFORM\COMMON\SRC\ARM\FREESCALE\MXARM11
WINCE500\PLATFORM\COMMON\SRC\ARM\FREESCALE\MX31
```

The PQOAL code is reusable:

- The PQOAL code in the `MXARM11` directory is reusable across all Freescale ARM11-based SOCs.
- The PQOAL code in the `MX31` directory is reusable across all platforms based on the i.MX31 device.

2.3 About the 3-Stack Platform Files

All of the driver and OAL content for the 3-Stack hardware platform is located in the directory:

```
WINCE500\PLATFORM\3DS
```

The `\3DS` platform directory implements the hardware abstraction routines invoked by the driver code in the Freescale CSP directory. In addition, this directory implements certain aspects of the PQOAL that may need to be modified by the OEM for their specific platform.

2.4 About the Sample Workspace

A default workspace is included in the BSP in the following directory:

```
\WINCE500\PBWorkspaces\3DSMobility
```

2.5 About the Support Files

Support files that complement the BSP source tree are located in the following directory:

```
\WINCE500\SUPPORT
```

This directory is used to store applications and tests targeted for the BSP. It also contains support files necessary to configure development tools and the HABToolKit for the 3-Stack board.

Chapter 3
Configuring OS Images

This chapter describes the configuration options available for the BSP. You can use Platform Builder to select one of the two default build configurations provided with the BSP sample workspace. These configurations control the type of OS image (debug, release, ship) that is generated by the Platform Builder tools.

In addition, the build configuration encapsulates all of the platform environment variables and custom build instructions that will be used during OS image creation.

NOTE

The sample workspace provided in the BSP is configured to generate a default image targeted for the i.MX31 3-Stack board hardware. It is not necessary to adjust any of the image configuration settings prior to building the BSP.

3.1 Selecting the Image Build Type

To choose the operating system image generated by Platform Builder, select **Set Active Configuration** on the Build OS menu. The sample workspace provided with the BSP provides two image build types:

Freescale 3DS: ARMV4I_Release—Retail build that includes KITL and kernel debugger support. This image type provides a smaller image with faster execution at the expense of limited debug capability.

Freescale 3DS: ARMV4I_Debug—Debug build that includes KITL and kernel debugger support. This image type provides full debug capability at the expense of a larger image size and slower execution.

For more information about the build types available for Windows CE 5.0, see the “Build Configurations” topic in the Platform Builder Help files.

3.2 Configuring the BSP Environment Variables

The BSP environment variables control the optional BSP support for OS images. Use the Platform Settings configuration dialog and the BSP Catalog to control the BSP environment variables. The sections that follow describe the methods of configuring BSP environment variables and provide a summary of the variables supported by the BSP.

3.2.1 Setting the Platform Environment Variables

Use the Platform Builder workspace to set the BSP environment variables for each build configuration.

To view and configure the variables, use these steps:

1. Open the sample workspace for the BSP.
2. From the Platform menu, choose **Settings**.
3. Select the Environment tab.

   The Platform Settings dialog (Figure 3-1) is displayed, showing a list of the environment variables. Table 3-1 describes the variables.

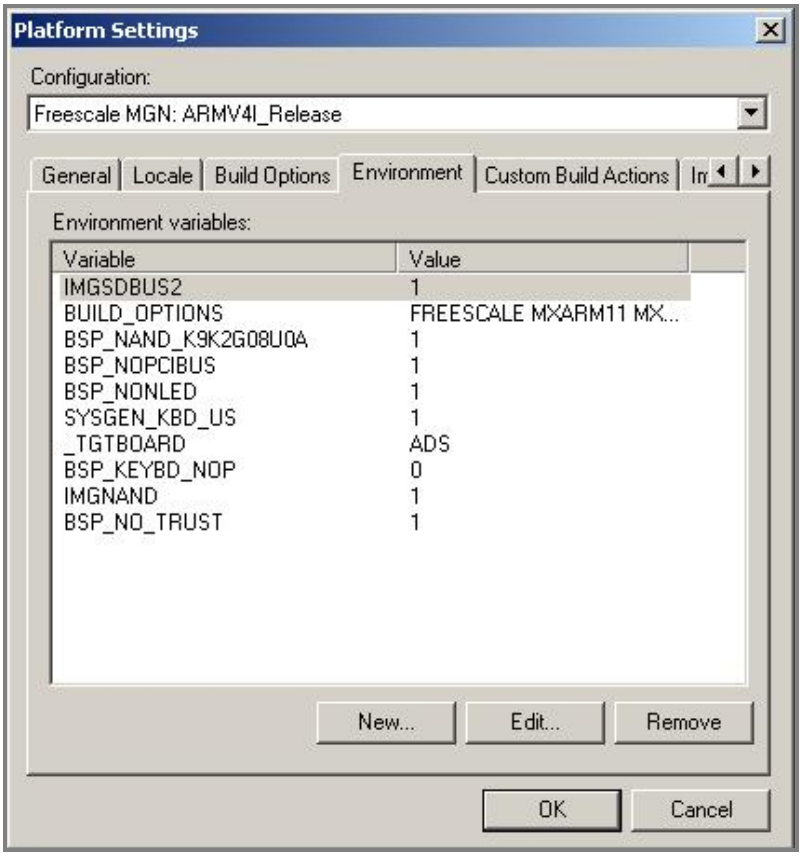

Figure 3-1 Platform Settings Dialog

Table 3-1 BSP Variables

BSP Variables	Description	Settings
BSP_NOPCIBUS	Used to exclude support for the PCI bus.	BSP_NOPCIBUS = 1 Excludes PCI bus support from the OS image. DO NOT remove from sample workspace.
BSP_NOCSPDDK	Used to exclude i.MX31 CSP driver development kit (CSPDDK) support. The CSPDDK is required for most BSP device drivers.	BSP_NOCSPDDK = 1 Excludes CSPDDK from the OS image. Use this configuration only when building an OS design that does not include BSP device drivers. DO NOT define for sample workspace.
BSP_NOAUDIO	Used to exclude support for audio driver.	BSP_NOAUDIO = 1 Excludes audio driver support from the OS image.
BSP_NODISPLAY	Used to exclude support for display driver.	BSP_NODISPLAY = 1 Excludes display driver support from the OS image.
BSP_NOTOUCH	Used to exclude support for touch driver.	BSP_NOTOUCH = 1 Excludes touch driver support from the OS image.
BSP_NONLED	Used to exclude support for notification LED driver.	BSP_NONLED = 1 Excludes notification NLED driver support from the OS image.
BSP_NO_TRUST	Excludes trusted environment support to the image.	BSP_NO_TRUST = 1 Excludes trusted environment support from the OS image. DO NOT remove from sample workspace.
_TGTBOARD	Allows conditional compilation for the 3-Stack hardware.	_TGTBOARD = ADS Configures the build for i.MX31 3-Stack hardware. **Note:** A clean build of all BSP components is required after modifying this setting.
BUILD_OPTIONS	Specifies an optional set of BSP directories for the build tools.	BUILD_OPTIONS = FREESCALE MXARM11 MX31 PMIC MGN Required for proper build of the BSP. DO NOT modify or remove from sample workspace.
IMGNAND	Used by EBOOT and OS build files to determine if images are targeted for NAND flash.	IMGNAND = 1 Link EBOOT and OS images for NAND flash. This environment variable is ignored for OS images if you have selected the following: **Platform Settings** > **Build Options** > **Write Run-time Image to Flash Memory** (IMGFLASH=1) to link the OS image for NOR flash.
BSP_USB_IRAM_PATCH	Use IRAM instead of DDR to store USB endpoint	BSP_USB_IRAM_PATCH = 1

Chapter 4
Building OS Images

This chapter explains how to build Windows CE OS images using the BSP.

4.1 Building CSP Libraries

The Freescale CSP libraries that support the i.MX31 are not generated during the **Build OS** > **Sysgen** build procedure. Microsoft Windows CE 5.0 ships with pre-built CSP libraries for various ARM processors, but the libraries for the i.MX31 are not included with the standard Microsoft distribution. Therefore, you must build the libraries for the i.MX31 separately.

To build the CSP libraries, use these steps:

1. Open the sample workspace.
2. Select the desired build type, as discussed in Section 3.1, Selecting the Image Build Type.
3. In the Platform Builder Workspace window, select the File View tab.
4. Expand the Favorites group to view the BSP sub-items.
5. Right-click on the FREESCALE CSP folder and select **Clean Before Building**.

6. Right-click on the FREESCALE CSP folder again and select **Build Current Project** (Figure 4-1).

For a release build type, the CSP libraries will be placed in:

```
\WINCE500\PUBLIC\COMMON\OAK\LIB\ARMV4I\RETAIL
```

For a debug build type, the CSP libraries will be placed in:

```
\WINCE500\PUBLIC\COMMON\OAK\LIB\ARMV4I\DEBUG
```

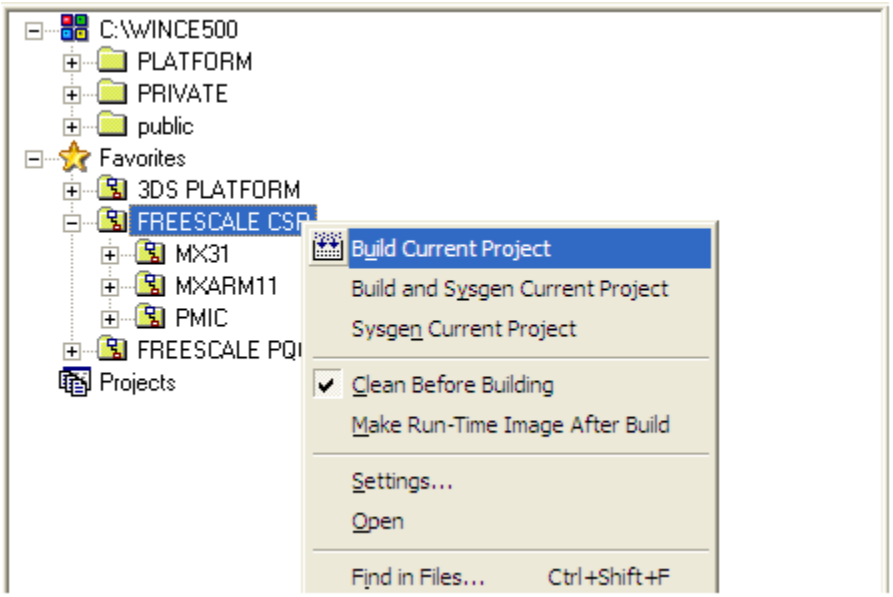

Figure 4-1 Building CSP Libraries

Note

The CSP libraries' build requires that all the Windows CE 5.0 updates specified in the *i.MX31 PDK Windows CE 5.0 Release Notes* have been installed. If not, the build process will have errors.

4.2 Building Run-Time Images

After building the Freescale CSP libraries, build the OS image by following the procedures in the Platform Builder documentation. For information, see the "Building a Run-Time Image" topic in the Platform Builder Help.

4.2.1 Building the BSP for the First Time

To build the BSP for the first time, use these steps:

1. Open the sample workspace.
2. Select the desired build type, as discussed in Section 3.1, Selecting the Image Build Type.
3. Make sure the Freescale CSP libraries are available by following the instructions in Section 4.1, Building CSP Libraries.
4. Using the **Build OS** menu, configure the build options as follows (Figure 4-2):
   - **Clean Before Building** unselected
   - **Copy Files to Release Directory After Build** selected
   - **Make Run-Time Image After Build** selected

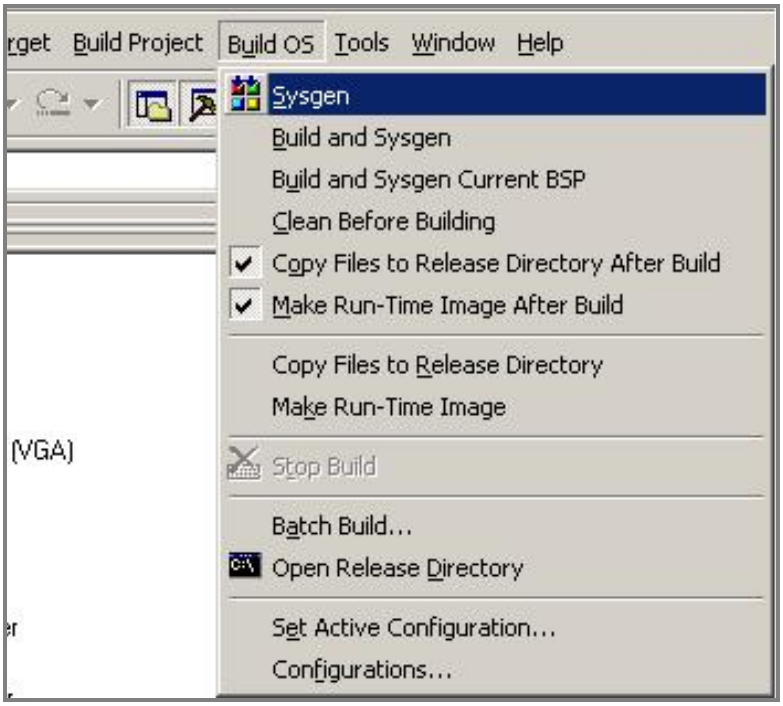

Figure 4-2 Building the BSP for the First Time

5. Select **Build OS** > **Sysgen** to start the build process.

   For a release build type, the resulting OS image files will be placed in the following directory:

   `WINCE500\PBWorkspaces\3DSmobility\RelDir\MX31_ARMV4I_Release`

   For a debug build type, the resulting OS image files will be placed in the following directory: `WINCE500\PBWorkspaces\3DSmobility\RelDir\MX31•ARMV4I•Debug`

4.2.2 Obtaining a Clean Build for the BSP

By default, Platform Builder performs incremental builds of the BSP components, even during the **Sysgen** build procedure. It may be desirable under certain circumstances to force a clean build for the BSP.

To obtain a clean build of all the BSP components, use these steps:

1. Follow the instructions in Section 4.1 Building CSP Libraries, to perform a clean build of the Freescale CSP libraries.
2. Using the Build OS menu, configure the build options as follows.
   - **Clean Before Building** selected
   - **Copy Files to Release Directory After Build** selected
   - **Make Run-Time Image After Build** cleared
3. Select **Build OS > Sysgen** to build the OS design and perform a clean build of the PQOAL libraries (Figure 4-3).

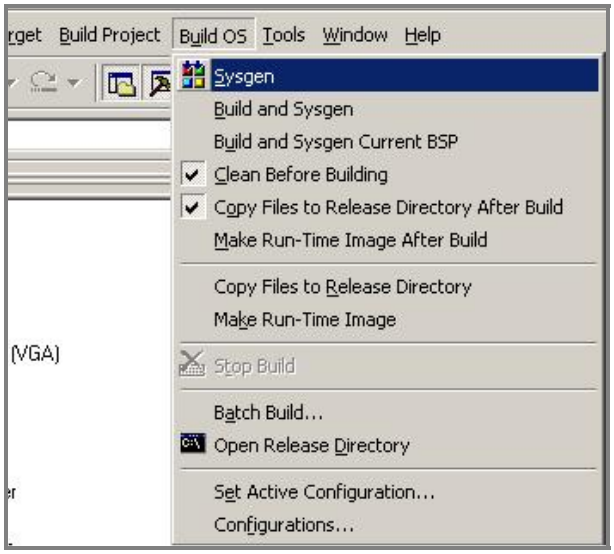

Figure 4-3 Sysgen a Clean OS Image

4. Using the Build OS menu, configure the build options as follows:
   - **Clean Before Building** selected
   - **Copy Files to Release Directory After Build** selected
   - **Make Run-Time Image After Build** selected

5. Select **Build OS > Build and Sysgen Current BSP** to perform a clean build of the BSP platform directory and complete the creation of an OS image (Figure 4-4).

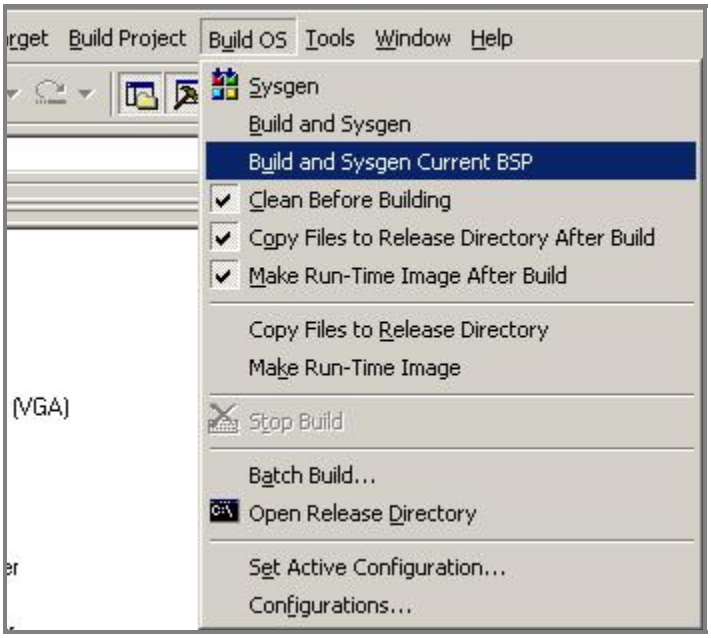

Figure 4-4 Build and Sysgen a Clean OS Image

4.2.3 Performing an Incremental BSP Build

The **Sysgen** build phase results in pre-built OS component binaries being copied to the release directory for the current build configuration. It is not necessary to perform a **Sysgen** again unless components are being added or removed from your OS design. Instead, you can perform an incremental build of the BSP components to quickly build an updated OS image.

To build an updated OS Image, use these steps:

1. Open a workspace.
2. Select the desired build type, as discussed in Section 3.1, Selecting the Image Build Type.
3. In the Platform Builder Workspace, select the File View.
4. Expand the Favorites group to view the BSP sub-items.
5. Right-click on any Favorites group folder and clear **Clean Before Building**.
6. If any Freescale CSP source files have been modified, perform an incremental build of the CSP libraries. To do so, right-click on the FREESCALE CSP folder and select **Build Current Project**.

7. If any PQOAL source files have been modified, perform an incremental build of the PQOAL libraries. To do so, right-click on the FREESCALE PQOAL folder and select **Build Current Project** (Figure 4-5).

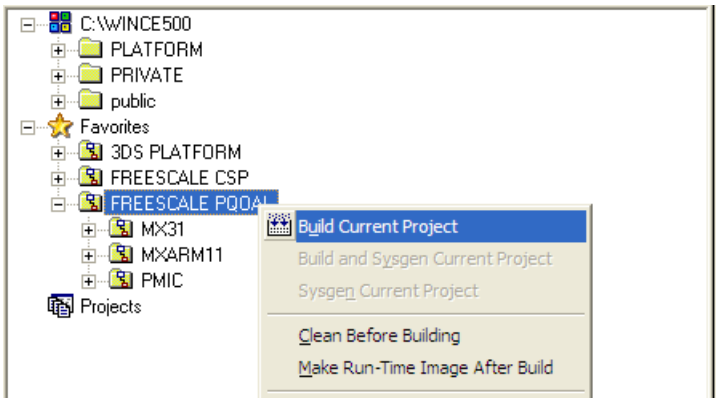

Figure 4-5 Incremental Build for CSP and PQOAL Libraries

8. Click **Build OS**, and then configure the build options as follows:
   - **Clean Before Building** unselected
   - **Copy Files to Release Directory After Build** selected
   - **Make Run-Time Image After Build** selected

9. Select **Build OS > Build and Sysgen Current BSP** to perform an incremental build of the BSP platform directory and complete the creation of an OS image (Figure 4-6).

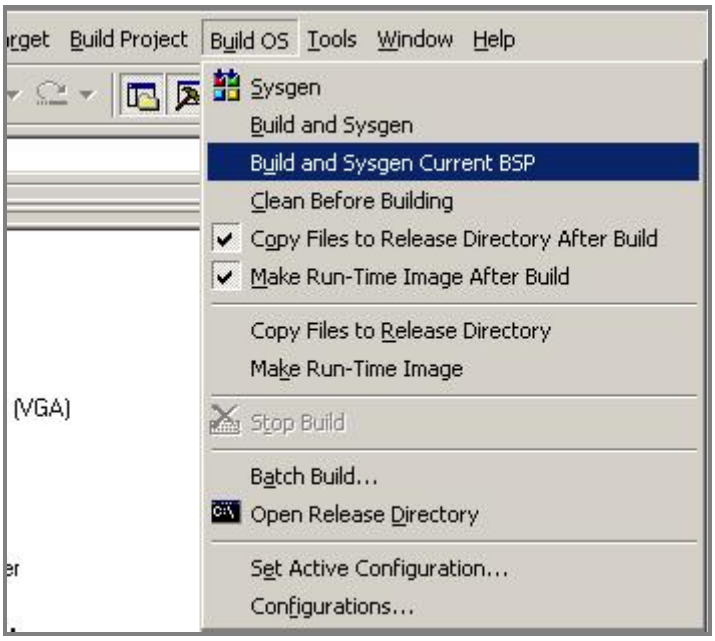

Figure 4-6 Incremental OS Image Build

4.3 Building an XLDR Image

The XLDR is a NAND boot loader that resides in the first 2K bytes of the external NAND device. The XLDR is the first code to execute on the system when booting directly from external NAND flash memory. The NAND flash controller copies the XLDR from the NAND memory to the internal NAND buffer after reset.

After the XLDR is copied, the CPU program counter is set to the base of the NAND buffer and begins executing the XLDR. The XLDR then loads a secondary boot loader (such as EBOOT, Windows Mobile IPL, and so forth), which can provide additional capabilities for booting OS images.

Once the OS Image is created, the XLDR image can be generated.

To build an XLDR image NAND for flash, use these steps:

1. In the Platform Builder Workspace, select the File View tab.
2. Expand the Favorites group, and then click **3DS Platform** > **SRC** > **Bootloader.**
3. Right-click on the XLDR folder and select **Clean Before Building**.
4. Right-click on the XLDR folder again and select **Build Current Project**, as shown in Figure 4-7). The XLDR binary image will be created in the workspace release directory.

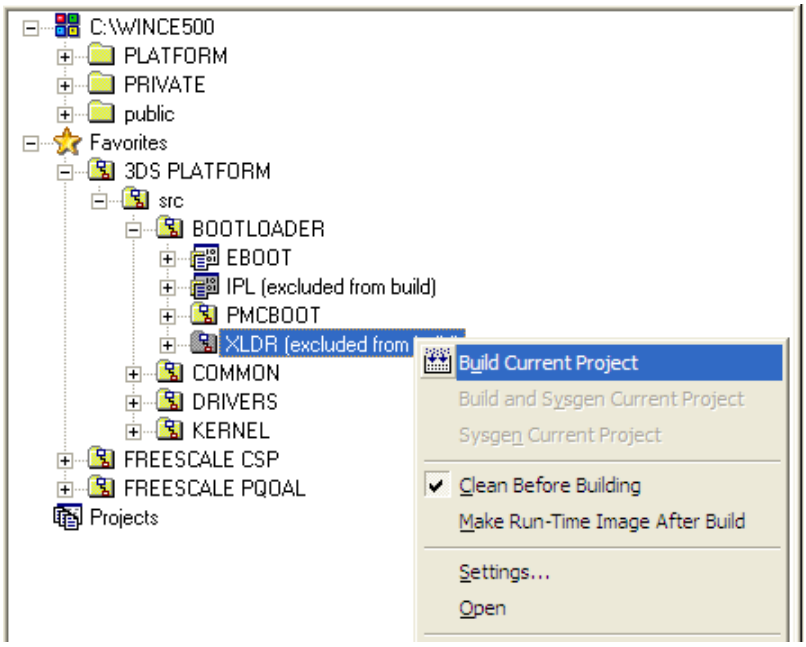

Figure 4-7 Building an XLDR Image

Chapter 5
Preparing for Downloading and Debugging

This chapter explains how to configure and initialize the target and development workstations, which you must do prior to downloading the OS images.

5.1 Configuring Boot Mode

Note

Switches SW5 through SW10 on the 3-Stack Debug board are used for Boot mode selection.

The following table shows jumper settings for the NAND Flash boot configuration.

Jumper Setting	Configuration
SW6	1
SW5-SW7-SW8-SW9-SW10	0

The following table shows jumper settings for the UART/USB boot configuration.

Jumper Setting	Configuration
SW5-SW6-SW7-SW8-SW9-SW10	0

5.2 Installing the Advanced ToolKit Package

You may obtain the Advanced ToolKit (ATK) package in either of two ways: from the i.MX31 PDK CD-ROM or as a free download from the Freescale website. For website information, see your Freescale representative.

To install the ATK, follow these steps:

1. Extract the contents from the `FSL_ATK_TOOL_WINS_STD_INSTALL.zip` file and place the files in a temporary folder.
2. From the top level folder, run the `setup.exe` file.

3. Follow the installation wizard to install the ATK on your PC.

   After installation, to access the ATK tool, click **Start** > **Programs** > **Advanced ToolKit** > **AdvancedToolKit V1.53** (Figure 5-1).

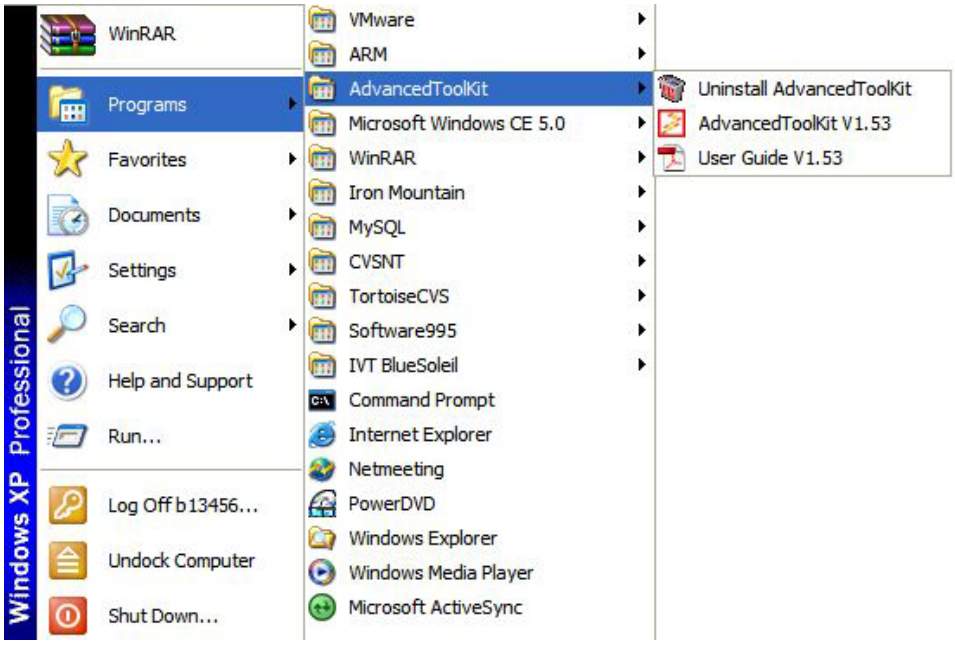

Figure 5-1 ATK Toolkit Location

5.3 Programming XLDR and EBOOT Using the ATK Tool

Certain users may receive a 3-Stack board without XLDR and EBOOT, and all users may at some time need to update those programs.

To program the XLDR, you should use the ATK tool, which supports K9F2G08R0A. The ATK tool is located in the Windows CE PDK package.

NOTE

Before flashing the files, you MUST erase the memory flash first.

5.3.1 Erasing Memory Flash

To erase memory flash, use these steps:

1. Set the boot mode (BOOTSTRAP No.4 pin configured as 0), in order to cause the MCU to boot from the UART/USB.

2. Power on or reset the device board.

3. Run `ADSToolkit.exe`.

   The Advanced ToolKit Configuration screen is displayed (Figure 5-2).

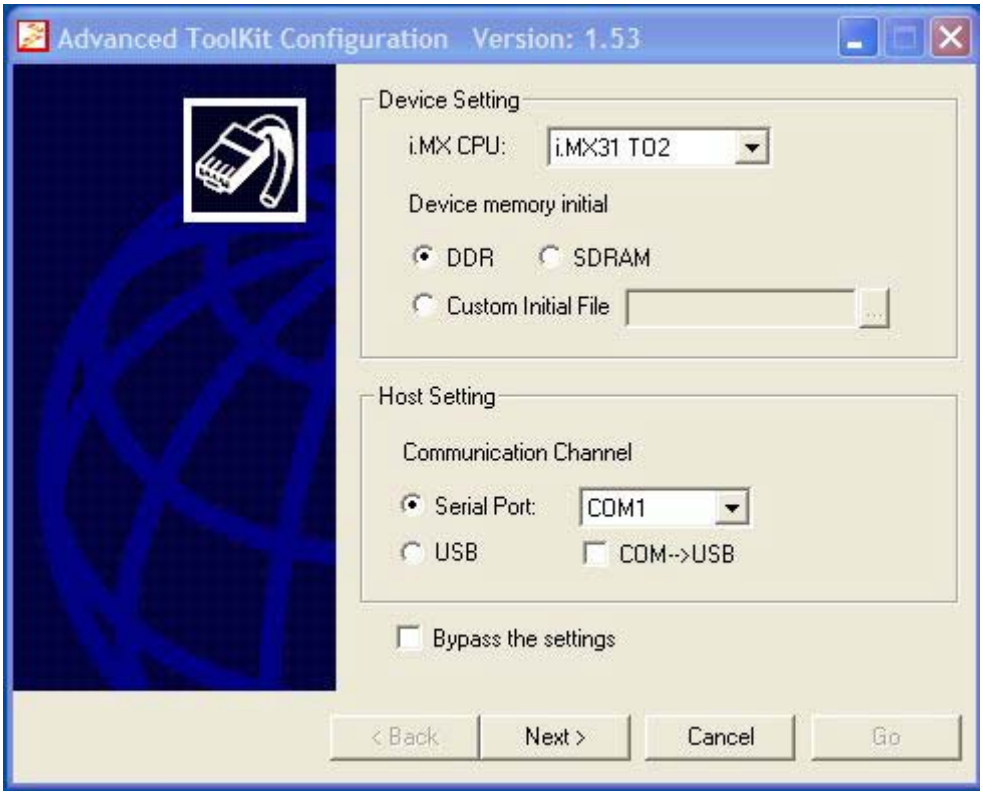

Figure 5-2 ATK Initial Configuration

4. Select these options:
   - For **i.MX CPU**, select **i.MX31_TO2**.
   - For **Device memory initial**, select **DDR**.
   - For **Communication Channel of Host Setting**, select the appropriate COM port/USB.
5. Click **Next**.

   The Advanced ToolKit Configuration dialog is displayed.
6. Click **Flash Tool**, and then click **Go**.

   The Flash Tool screen is displayed.

7. Select the options shown in Figure 5-3, and then click **Erase** to erase the flash.

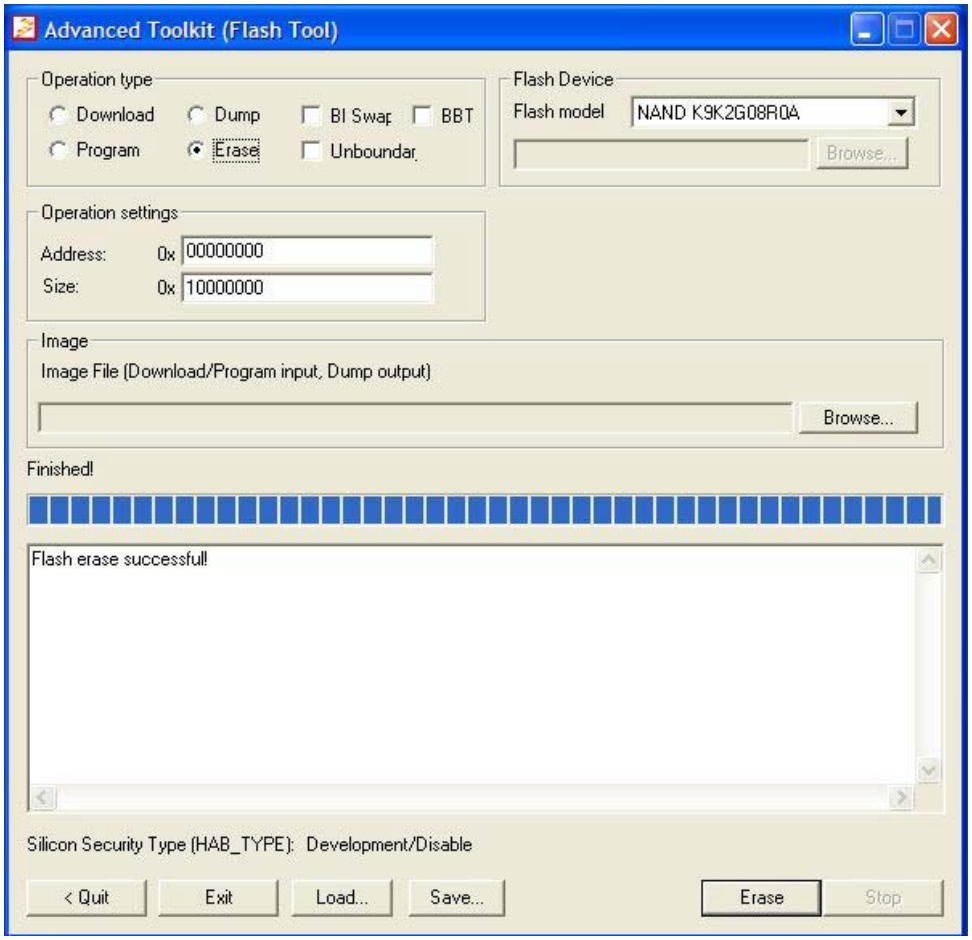

Figure 5-3 Erase Flash with ATK

Note

The NAND flash device used in the 3-Stack board is **K9F2G08R0A**, but the **K9K2G08R0A** from the ATK Flash model options works with this memory.

5.3.2 Program XLDR

To program the XLDR, use these steps:

1. In the **Image** section, click **Browse** to select **XLDR.nb0** as the Image File.
2. Select the options shown in Figure 5-4, and then click **Program** and wait until flashing is successful.

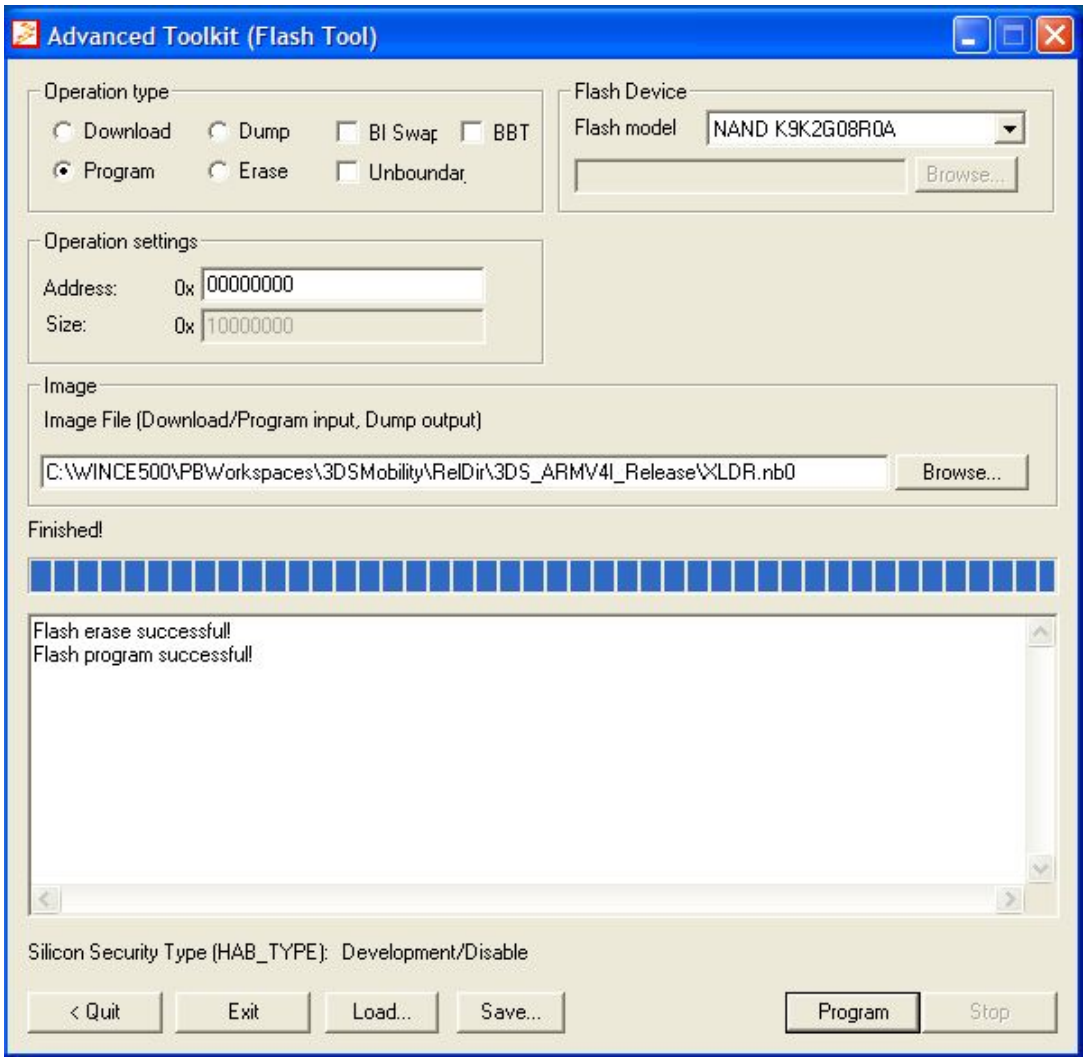

Figure 5-4 XLDR.nb0 Programming

5.3.3 Programming EBOOT

To program **EBOOT**, use these steps:

1. In the **Image** section, click **Browse** to select **EBOOT.nb0** as the Image File.
2. Select the options shown in Figure 5-5, and then click **Program** to write EBOOT to the board.

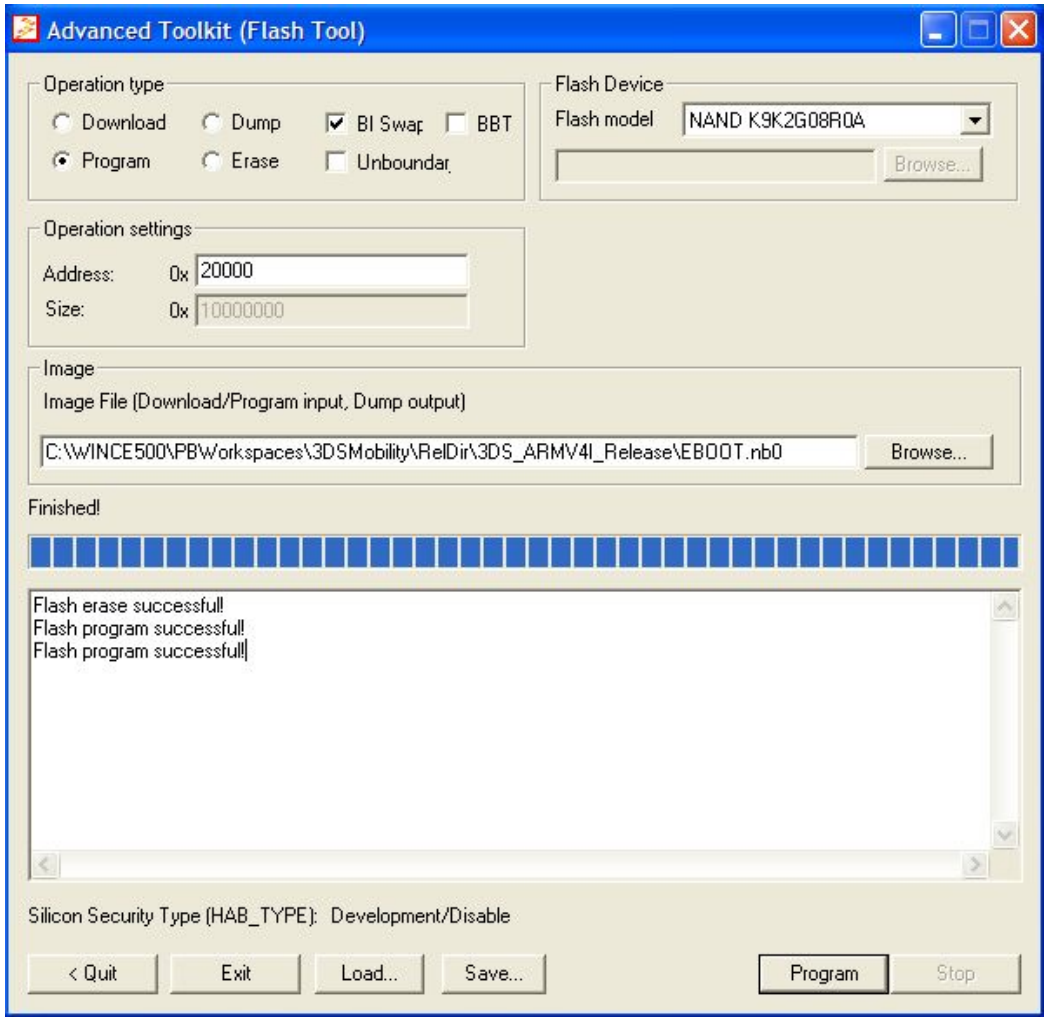

Figure 5-5 EBOOT.nb0 Programming

3. Set the board for NAND flash boot, using the instructions in Section 5.1, Configuring Boot Mode.
4. Power on or reset the device board.

5.4 Programming the OS Kernel Image

Use Platform Builder and EBOOT to program the OS kernel image into NAND Flash. When you have built the OS Image and loaded the XLDR and EBOOT into the NAND Flash, change the Boot Settings to NAND Flash Boot, and then use the following steps to program the OS image to the board.

Note

Before building the OS image, ensure that the environment variable IMGNAND = 1 in Platform Builder is set. If not, set it by selecting **Platform** > **Settings** > **Environment**.

5.4.1 Adding the 3-Stack Board to Platform Builder

To add the 3-Stack board to the Platform Builder, use these steps:

1. Connect the 3-Stack board to the network with a standard UTP cable.
2. Connect the 3-Stack board to your host PC using a serial cable.
3. Open a serial console on your host PC with the following settings.
   - Baud Rate: 115200
   - Data bits: 8
   - Parity: none
   - Stop bits: 1
   - Flow Control: none
4. Ensure that the board is configured to boot from the NAND flash (see Section 5.1, Configuring Boot Mode).
5. Power up the 3-Stack board (the EBOOT startup sequence should appear in the serial console, as shown in Figure 5-1). If the EBOOT startup sequence is not displayed, program XLDR and EBOOT following the procedure in Section 5.3, Programming XLDR and EBOOT Using the ATK Tool.

6. Press the spacebar on the host PC to enter the EBOOT menu before the three-second period ends, and then configure the ETH settings to match your network configuration (IP address, Subnet mask, MAC address, and DHCP options; see Figure 5-6 and Figure 5-7).

If your networks include a DHCP server, ensure that option **3 DHCP** is set to **Enabled**.

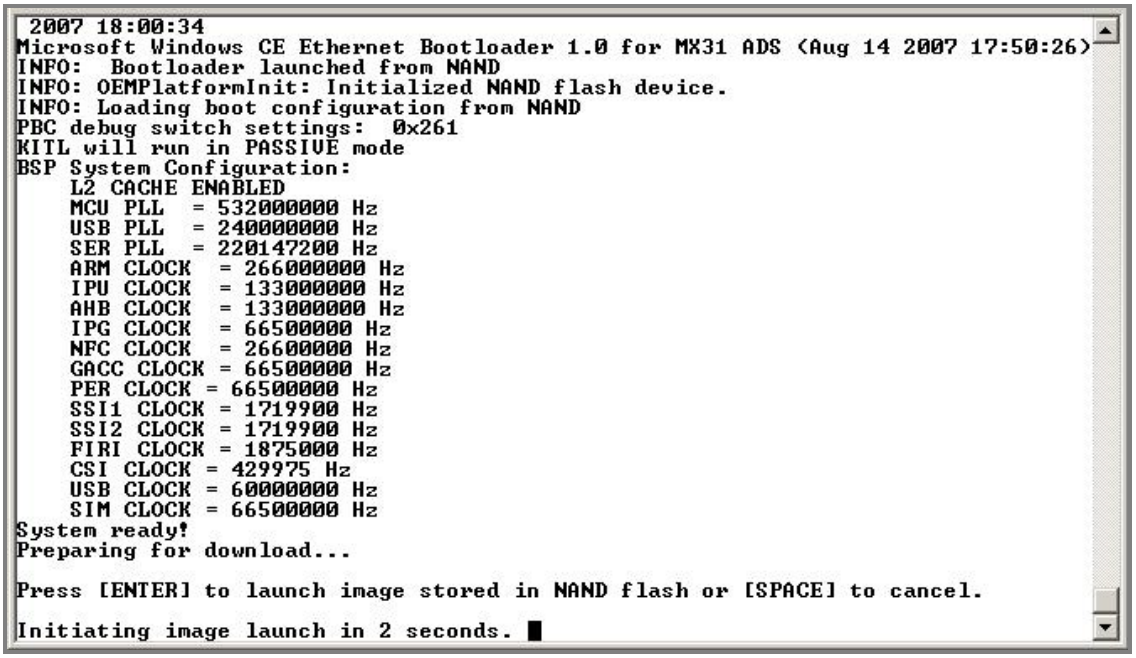

Figure 5-6 EBOOT Startup Sequence

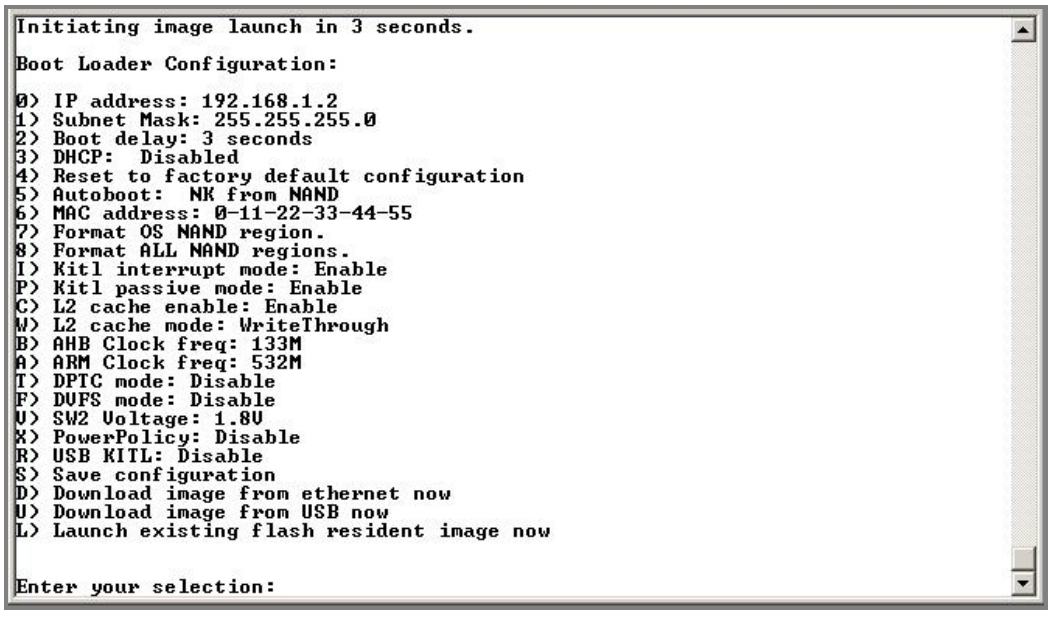

Figure 5-7 EBOOT Menu with ETH Configuration for Direct Connection

7. In Platform Builder, select **Target** > **Connectivity Options** (Figure 5-8).

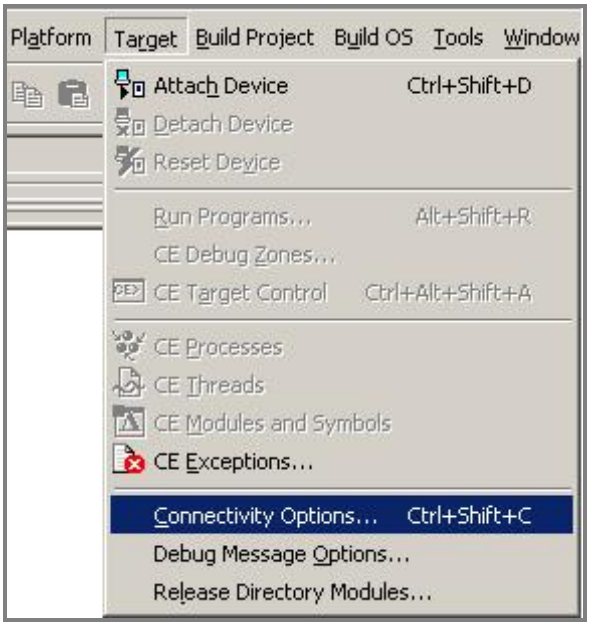

Figure 5-8 Platform Builder Connectivity

The Target Connectivity dialog is displayed (Figure 5-10).

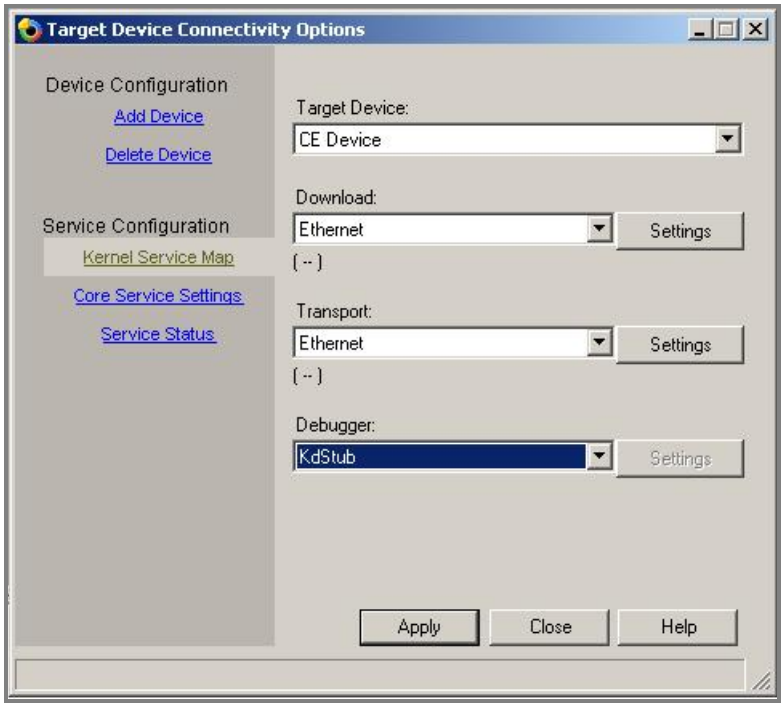

Figure 5-9 Target Connectivity Options

8. Next to **Transport**, click the down arrow and select **Ethernet**.
9. Next to **Debugger**, click the down arrow and select **KdStub**.
10. Next to **Download**, click **Settings**.

    The Ethernet Download Settings window is displayed.

11. On the EBOOT menu, press "**D**" to request the download of the image to the Platform Builder.

    The download begins, as shown in Figure 5-10.

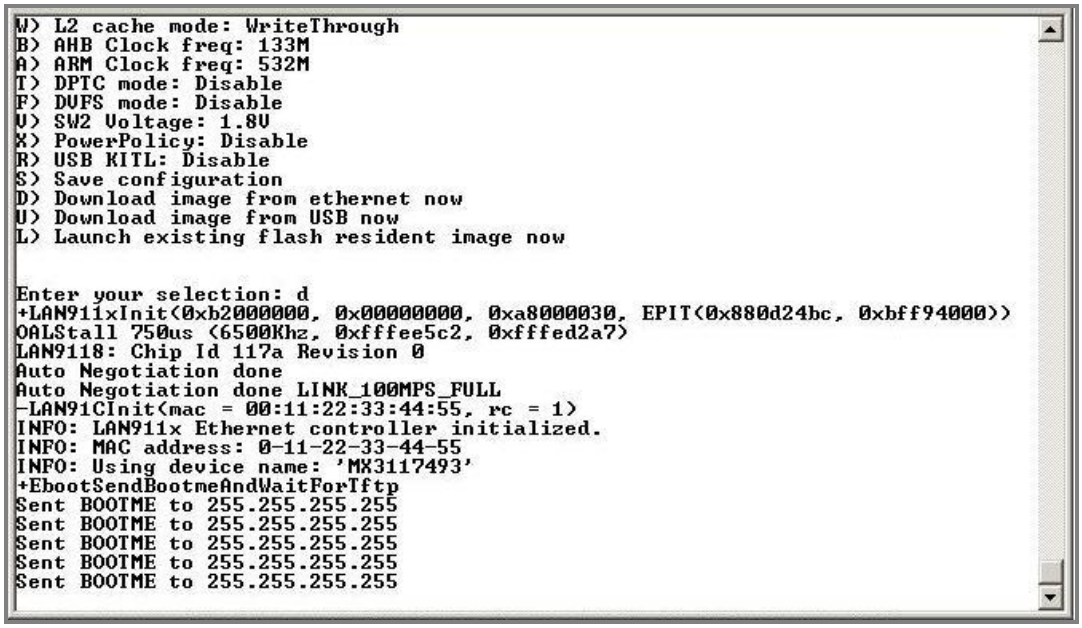

Figure 5-10 EBOOT Request Download Message Screen

Note

You may need to stop Black ICE software. To do so, in the Start menu, click **Settings** > **Control Panel > Administrative Tools > Services**. Then right-click on **Black ICE** and click **Stop**.

12. When the EBOOT program displays the BOOTME messages, the 3-Stack board device should be displayed in the Ethernet Download Settings dialog, as shown in Figure 5-11. Double-click the active device, and then click **OK**.

If the device is not displayed in the Active Devices field, check that the ETH settings are properly configured and that there are no security software blocking broadcast signals (such as Black ICE software) on the host PC.

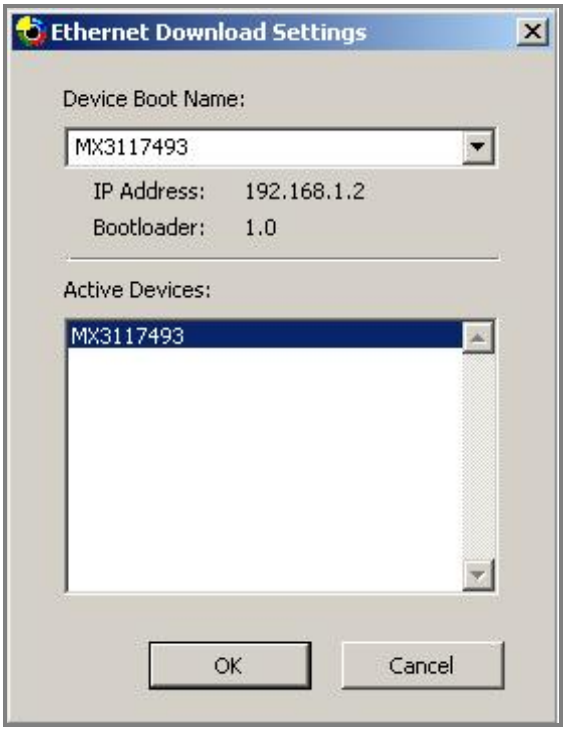

Figure 5-11 Ethernet Download Settings

The Target Device Connectivity Options screen is displayed. The Target Device box should display **CE Device** (Figure 5-12).

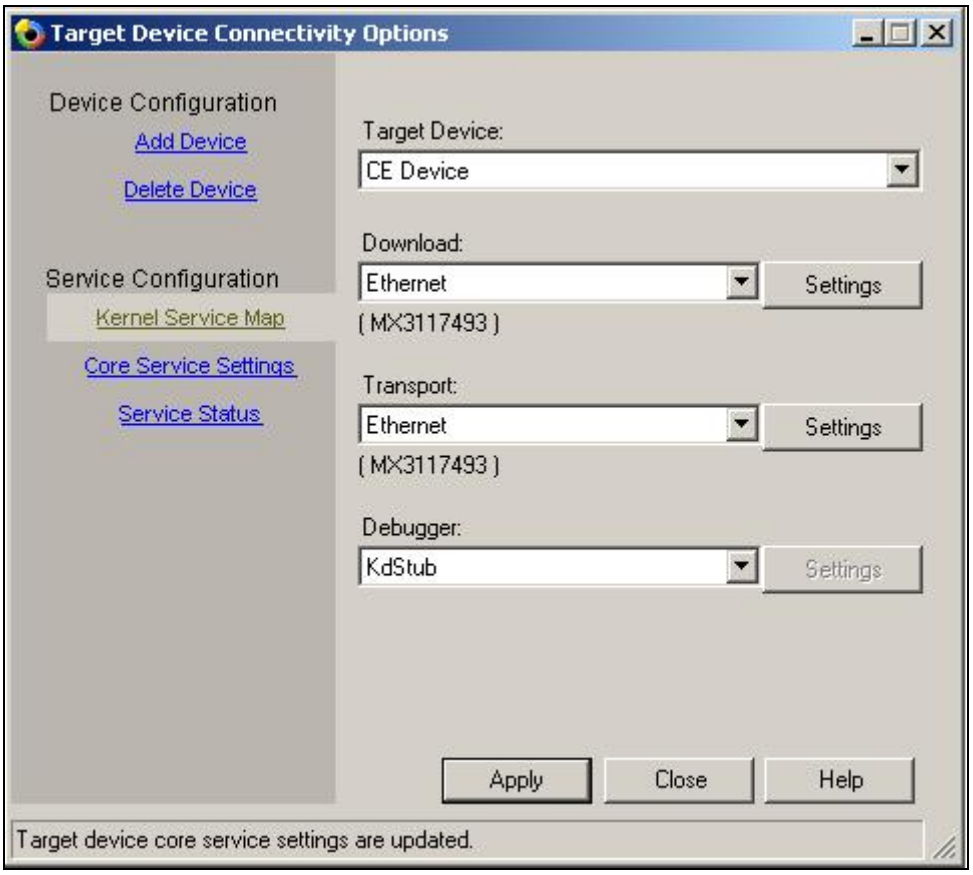

Figure 5-12 Target Device Connectivity Settings

13. Click **Apply**, and then click **Close**.

5.4.2 Downloading the OS Kernel Image using EBOOT

To download the OS kernel image using EBOOT, use these steps:

1. Click **Target** > **Attach Device** (see Figure 5-13).

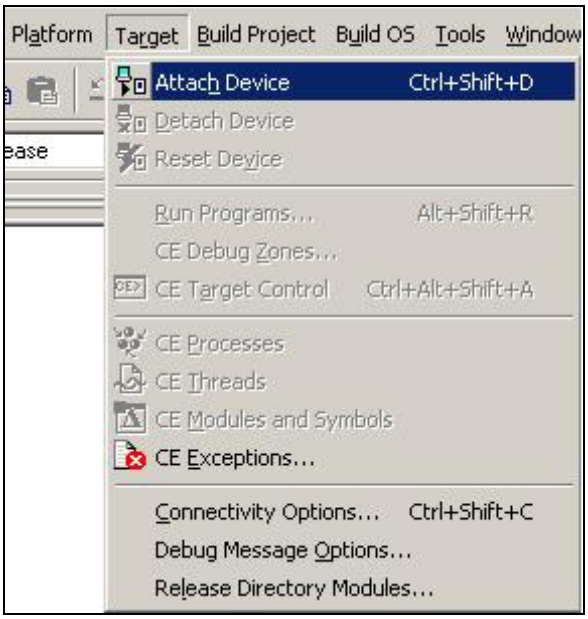

Figure 5-13 Attach Device to Download OS Image

The EBOOT menu is displayed.

2. Press "**D**" to begin sending the BOOT ME messages to download the OS Image from the Platform Builder.

   The Download Runtime Image to CE Device dialog is displayed, showing the progress of the download (Figure 5-14).

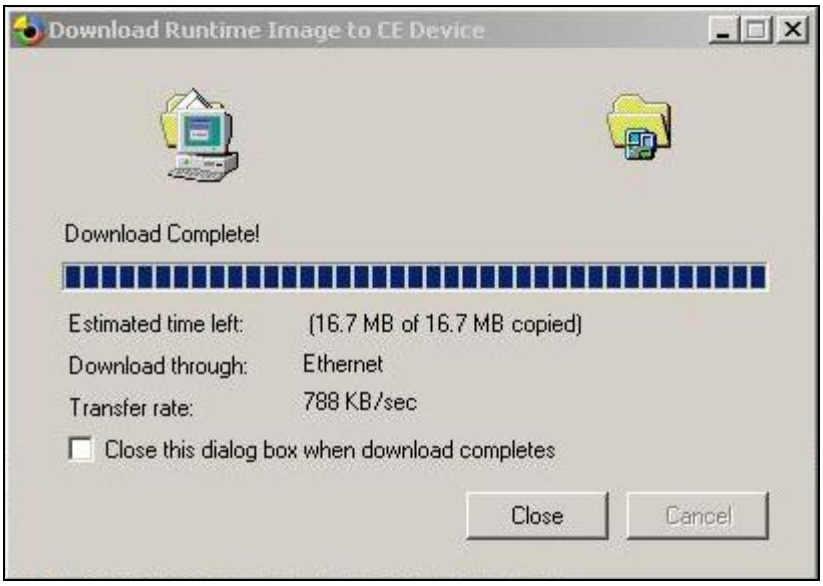

Figure 5-14 Download Progress

3. Wait until the Platform Builder finishes the OS Image download.

4. At the serial console, press the '**y**' key to begin programming the flash.

   EBOOT will program the image into flash and display the status using serial debug messages (Figure 5-15).

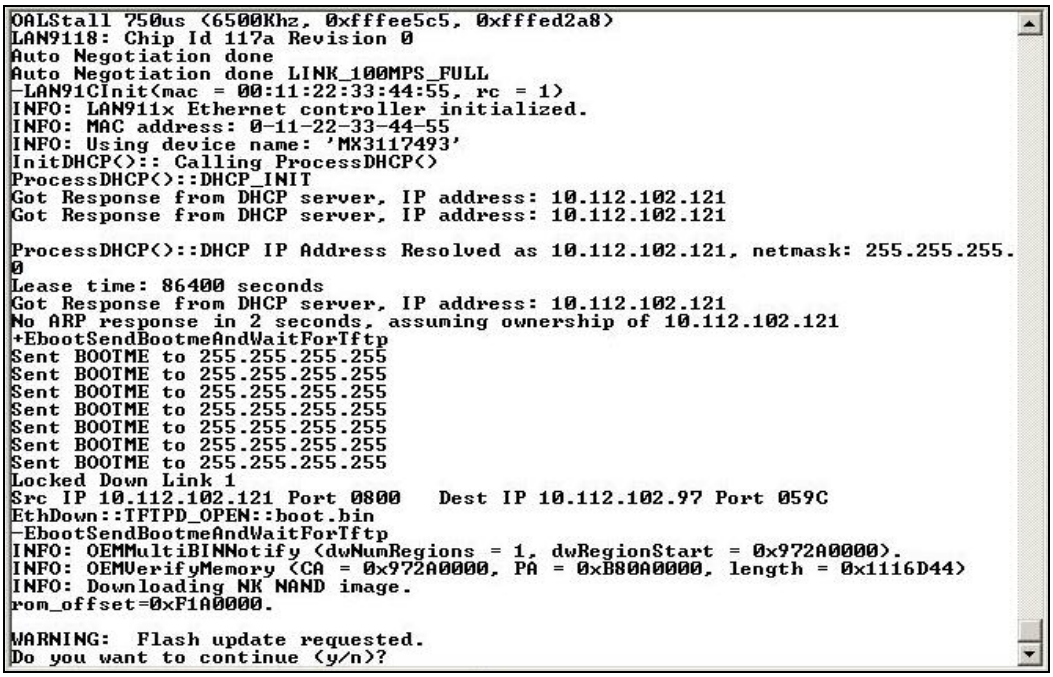

Figure 5-15 EBOOT Flash Update

5. Reset the board after flashing.
6. Enter the EBOOT menu at the serial console.
7. Set **Autoboot** to **NK from NAND** option from the EBOOT menu.
8. Save the settings using the "**S**" option on the EBOOT menu.

   Now the Windows CE system can automatically boot from NAND Flash.

5.5 Updating XLDR and EBOOT

You can use the ATK tool or Platform Builder to update XLDR and EBOOT.

5.5.1 Update using ATK Tool

To use the ATK tool to update XLDR and EBOOT directly, repeat the steps in Section 5.3, Programming XLDR and EBOOT Using the ATK Tool.

5.5.2 Update using Platform Builder

If XLDR and EBOOT have been previously installed in the 3-Stack board, you can use Platform Builder to update them.

XLDR

To update XLDR, use these steps:

1. Click **Platform** > **Settings**.
2. In the General section, select `XLDR.bin` as the file name for the run-time image.
3. In Platform Builder, download into the 3-Stack board using EBOOT.

Note

Do not use the **File** > **Open** ... method to open `XLDR.bin`. Doing so produces an error message stating that the file cannot be opened as a workspace because "it is not a valid Windows CE BIN file".

EBOOT

To update EBOOT, use these steps:

1. Open `PBWorkspaces\3DSmobility\RelDir\3DS_ARMV4I_Release\eboot.bin`
2. In Platform Builder, download into the 3-Stack board using EBOOT.

Figure 5-16 shows the EBOOT Flash update results.

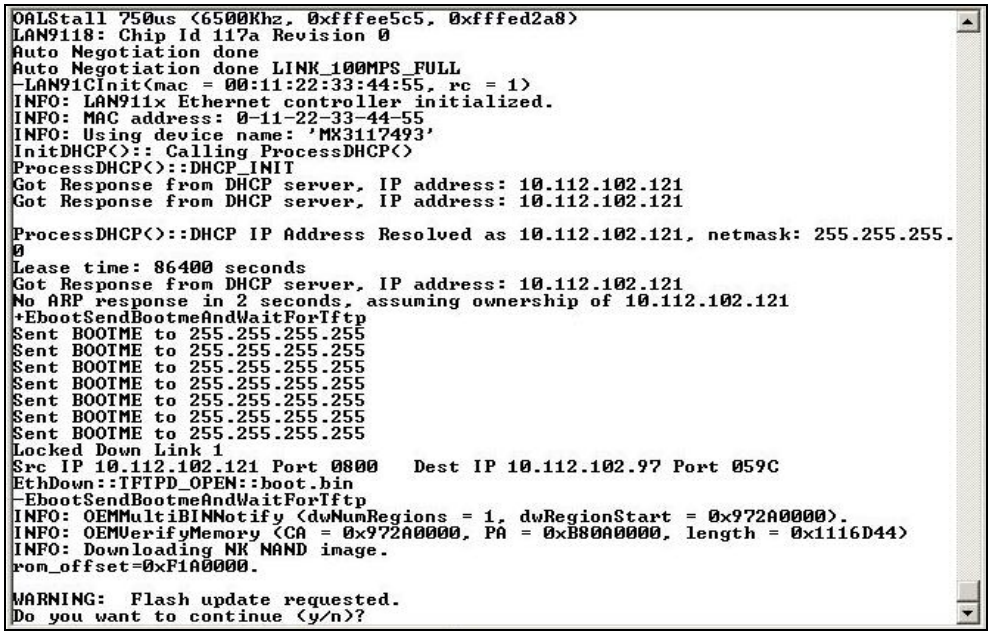

Figure 5-16 EBOOT Flash Update

5.6 Understanding the EBOOT Menu Options

Table 5-1 describes the commands on the EBOOT menu.

Table 5-1 EBOOT Menu Commands

Identifier	Item	Description
0	IP Address	Configures a static IP for the board in order to connect EBOOT with the Platform Builder. When the DHCP option is enabled this IP address is not used.
1	Subnet Mask	Configures the subnet mask for the board's static IP address.
2	Boot Delay	Configures the time given to execute the Autoboot option of EBOOT.
3	DHCP	Enables or Disables the DHCP client of EBOOT. This option allows the board to obtain a valid IP address from the network DHCP server to connect to Platform Builder.
4	Reset to factory default configuration	Restores the EBOOT default values.

5	Autoboot	Configures the boot option for the OS image. The options can be: • Disabled. At boot will enter the EBOOT menu. • NK from NOR. EBOOT loads the OS image from NOR flash (not used on 3-Stack board) • NK from NAND. EBOOT loads the OS image from NAND flash OS region.
6	MAC address	Configures the MAC address for the board's external Ethernet controller.
7	Format OS NAND Region	Formats the OS NAND region; erases any existing image in this region.
8	Format all OS NAND Regions	Formats NAND flash; erases all NAND regions existing programs, including EBOOT.
I	KITL Interrupt mode	Enables/Disables the KITL interrupt mode instead of the KITL polling mode.
P	KITL Passive mode	Enables/Disables KTIL's passive mode; this option bypasses KITL connection if not connection is established with Platform Builder
C	L2 Cache Enable	Enables/Disables the L2 cache from the ARM processor.
W	L2 Cache Mode	Configures the L2 cache mode to be Write Through or Write Back.
B	AHB Clock Frequency	Configures the AHB clock at 66.5 MHz or 133 MHz speed.
A	ARM Clock Frequency	Configures the ARM core clock at 532 MHz or 266 MHz speed.
T	DPTC Mode	Enables/Disables the Dynamic Process and Temperature Compensation feature for the BSP. For more information see the *MCIMX31 Reference Manual*.
F	DVFS Mode	Enables/Disables the Dynamic Voltage Frequency Scaling feature of the BSP. For more information see the *MCIMX31 Reference Manual*.
V	SW2 Voltage	This feature is not functional for this version of the BSP.
R	USB KITL	Enables/Disables the USB as KITL transport.
S	Save Configuration	Saves the current configuration of EBOOT.
D	Download Image from Ethernet Now	Downloads the OS Image or EBOOT image using the Ethernet connection.
U	Download Image from USB Now	Download an OS image or bootloader using USB RNDIS service.
L	Launch Existing Flash Resident Image Now	Launch the OS image stored in NAND.

5.7 Using UBOOT to Boot the Board

The UBOOT program is an alternative way to boot the board, by downloading images from Platform Builder to the board, using USB rather than Ethernet.

To use UBOOT, follow these steps:

1. If you want to use the USB KITL before building the kernel, remove all USB driver modules from OSDesignView in Platform Builder.
2. Open a serial terminal at the host PC.
3. Connect a serial cable and a USB cable between the board and the PC, and then apply power to the board.
4. At the EBOOT menu, change the boot configuration to match the following:

   0) IP address: 192.168.0.2

   1) Subnet Mask: 255.255.255.0

   3) DHCP: Disabled

   I) Kitl interrupt mode: Disable

   P) Kitl passive mode: Disable

   R) USB KITL: Enable

5. Configure your host PC network settings to be in the same network as the board.

Note

The IP address must be different from that of the current network range. Refer to 192.168.0.*x,* where *x* is between 1 and 254. Save this configuration.

6. Press the '**u**' key to download the image from the USB.

7. If you are using UBOOT or USB KITL for the first time, the PC will prompt you to install a driver (Figure 5-17).

   For information about installing the driver, see:

   `\WINCE500\PUBLIC\COMMON\OAK\DRIVERS\RNDISMINI\HOST\howto.txt`

Figure 5-17 Installing RNDIS Driver on Windows XP

8. Select **Install from a list or specific location (Advanced)**.

9. Click **Next**.

The Search and Installation screen is displayed (Figure 5-18).

Figure 5-18 Selecting a Driver for RNDIS

10. Select **Include this location in the search**, and then click the down arrow to add the following path:

`\WINCE500\PUBLIC\COMMON\OAK\DRIVERS\ETHDBG\RNDISMINI\HOST\`

This will use the `usb8023.inf` file to install the RNDIS driver for the board.

11. Click **Next**.

A pop-up window asks for the `usb8023.sys` file to install. This file is usually located at:

`\WINDOWS\system32\drivers folder for Windows XP`

12. For installation problems, see the following file:

```
CE500\PUBLIC\COMMON\OAK\DRIVERS\ETHDBG\RNDISMINI\HOST\ howto.txt
```

After you install the driver correctly, you will see the Microsoft Windows CE RNDIS virtual adapter from Network Connections displayed on the PC.

Note

It is important to configure this network connection properly. Use a static IP address, and then reboot the board. Here is an example of the static IP address:

IP address: 192.168.0.4 //same subnet, different IP address from that used on the board.

Subnet mask: 255.255.255.0 //must be the same as the boot configuration on the board.

Default Gateway: 192.168.0.1

Note

To add the device to Platform Builder, use the steps in Section 5.4.1 Adding the 3-Stack Board to Platform Builder. At step 11 select the "**U**" option instead of "**D**" to use the USB transport instead of the Ethernet transport.

Chapter 6
Windows CE Applications in the BSP

This BSP supports the following Windows CE 5.0 applications: Camera, TV-Out, Rotate, FM Radio, 3D Demo, Image Update, and Image Burner.

6.1 Camera Application

The camera application demonstrates the capabilities of the camera driver supplied with the BSP. You can use this application to manipulate the Omnivision OV2640 Camera CMOS sensor, using the 3-Stack camera driver.

6.1.1 Building a Camera Application

To build a Camera application, use these steps:

1. Create an **APP** folder under the following folder:
   `\Platform\3DS\Src`
2. Create an empty **dirs** file under the following folder:
   `\Platform\3DS\Src\App`
3. Copy the **CAMAPP** folder from the `\Support\App` folder to the `\Platform\3DS\Src\App` folder.
4. Select the File View tab of the Platform Builder Workspace window.
5. Expand **Favorites** > **3DS Platform** > **src** > **App**.
6. Right-click on the **CAMAPP** folder and select **Clean Before Building**.
7. Right-click on the **CAMAPP** folder again and select **Build Current Project**.

   The CAMAPP execution file will be created in the workspace release directory.

6.1.2 Using the Camera Application

Table 6-1 describes the features and commands for the camera demonstration application.

Table 6-1 Camera Demo Application Features

Feature	Description
Picture Browser	**File** > **Picture browser** **View** (button at the bottom of the application) The picture browser is used for viewing a picture.
Picture Zoom	**View** > **Zoom** Only supports 1, 2 Zoom. (Ovt2640 sensor does not support zoom.)
Change Resolution	**View** > **Resolutions** Can switch output resolution among the following QQVGA, QCIF, QVGA, CIF, and VGA.
Horizontal Flipping	**View** > **Horizontal Flipping** For flipping picture horizontally.
Vertical Flipping	**View** > **Vertical Flipping** For flipping picture vertically.
Options	**View** > **Options** Change Resolution, Zoom, and Rotation in one command.
Capture a still picture	**Capture** (button at the bottom of the application) The captured picture is the same resolution that that you previously set in "**View** > **Resolution**"
Save or discard a still picture	1. Press **Capture** (button at the bottom of the application). The name of the button name changes to **Save** or **Discard**. 2. Press **Save** to save the picture. You may define the saved file name and type (*.jpg ,*.bmp). 3. Press **Discard** to discard the picture.
Rotation	Press **Rotate** (button at the bottom of the application). The image rotates 90 degrees.
Frame rate display	Displays the current frame rate at the lower right of the application.

6.2 TV-Out Application

The TV-Out application is located at `\windows\TVOUT.exe`. The TV-Out application toggles the Windows CE desktop between the LCD display and the TV-Out connector.

TV-Out supports PAL and NTSC standards. The default mode for TV-Out is PAL; to change the mode, call the TV-Out program with a mode parameter.

- `tvout.exe 0` sets the TV output format to PAL and sends the Windows CE desktop to the TV-Out connector.
- `tvout.exe 1` sets the TV output format to NTSC and sends the Windows CE desktop to the TV-Out connector.

You can execute `tvout.exe` in either of two ways:

- Use the **Start** > **Run**...window.
- Use the default output format (PAL) by double-clicking on the `tvout.exe` program. When the TV is the display screen, double-clicking this file toggles the display to the LCD.

Note

Before running the TV-Out application, close all Windows.

6.3 Rotate Application

The Rotate application is located at `\windows\rotate.exe`. The Rotate application switches the display between Portrait and Landscape modes. Landscape Mode has higher performance, but most settings in Windows CE are designed to use Portrait Mode.

- When the display is in Portrait Mode, double-clicking this file toggles the display to Landscape Mode.
- When the display is in Landscape Mode, double-clicking this file toggles the display to Portrait Mode.

Note

The Rotate application is designed to configure the Window settings only in Portrait Mode. Rotate does not support all applications in both Portrait and Landscape Mode.

6.4 FM Radio Application

The FM Radio application enables a user to listen to an FM radio station. This PDK uses the Si4702 chip as an FM radio receiver, and provides the FM radio driver. The sound is sent to the audio output.

6.4.1 Building the FM Radio Application

To build the FM radio application, use these steps:

1. Create an **APP** folder under the following folder:
   `\Platform\3DS\Src`
2. Create an empty **dirs** file under the following folder:
   `\Platform\3DS\Src\App`
3. Copy the **FMRadioApp** folder from the `\Support\App` folder to the `\Platform\3DS\Src\App` folder.
4. Select the File View tab of the Platform Builder Workspace window.
5. Expand **Favorites** > **3DS Platform** > **src** > **App**.
6. Right-click on the FMRadioApp folder, and then select **Clean Before Building**.
7. Right-click on the FMRadioApp folder again, and then select **Build Current Project**.

   The **FMRadioApp** execution file will be created in the workspace release directory.

6.4.2 Using the FM Radio Application

To use the application, click the FM Radio Application icon.

Figure 6-1 illustrates the FM Radio's main window. The application window will fit the new client area dimensions after screen rotation, such as from Portrait Screen to Landscape Screen. Table 6-2 describes the Window options. Figure 6-2 through Figure 6-5 display the secondary dialogs.

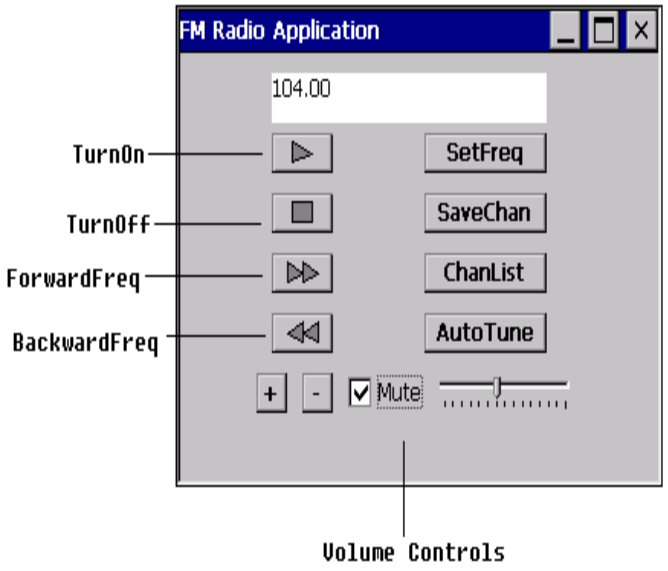

Figure 6-1 FM Radio Application Main Window

Table 6-2 FM Radio Application Main Window Controls

Control	Action
Turn On	Plays the displayed frequency.
Turn Off	Stops playing the FM radio.
ForwardFreq	Increases the frequency in increments of 0.10 MHz.
BackwardFreq	Decreases the frequency in increments of 0.10 MHz.
Volume controls	Sets the FM radio volume to the specified value. The volume value is shown on the volume track. Valid settings are 0 (lowest) to 15 (loudest). • Click **+** to increase the volume. • Click **-** to decrease the volume • Click **Mute** to silence the radio audio.
SetFreq	Displays the Set Frequency dialog. See Figure 6-3. Displays the current frequency. To

	change the frequency setting, enter the desired frequency, and then click **OK**.
SaveChan	Displays the Save Channel Name dialog. See Figure 6-4. To create a name for the channel of the selected frequency, enter a channel name, and then click **OK**. The name and frequency are saved to a channel list.
ChanList	Displays the saved channel list. See Figure 6-5. You may add, edit or delete a channel. To play a channel, select the channel and click **OK**.
AutoTune	Performs automatic tuning and saves the scanned channels to a selected channel list. Click **ChanList** dialog to check the scanned channels.

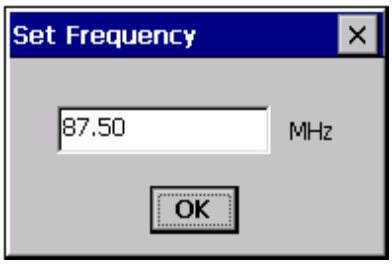

Figure 6-2 Set Frequency Dialog

Figure 6-3 Save Channel Name Dialog

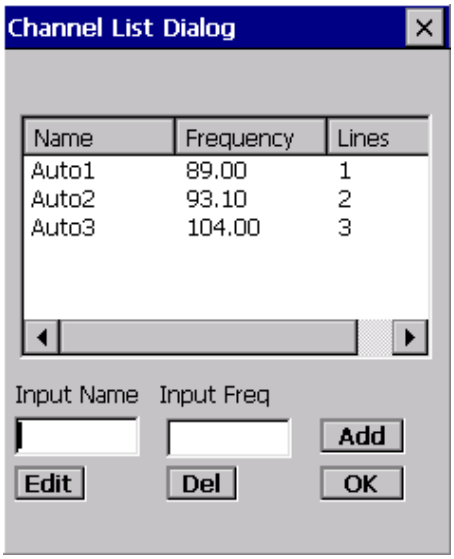

Figure 6-4 Channel List Dialog (Portrait Screen)

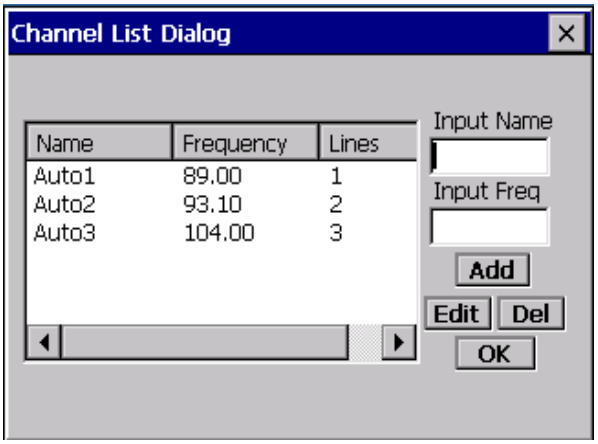

Figure 6-5 Channel List Dialog (Landscape Screen)

6.5 3D Demo Application

Two applications demonstrate the MBX driver 3D feature:

- D3DM: Located at `\windows\D3DMEvilSkull.exe`, `D3DMEvilSkull.exe` uses the Microsoft D3DM API. Double-clicking this file displays geometry morphing.
- OPENGLES: Located at `\windows\OGLESVase.exe`, `OGLESVase.exe` uses the Khronos OPENGLES API. Double-clicking this file displays texturing with transparency and environment mappings.

To run the 3D demonstration applications, you must add MBX catalogs. For instructions on adding these catalogs, see Chapter 18 "MBX Direct3D Mobile/OpenGL ES Drivers" in the *i.MX31 PDK Windows CE 5.0 Reference Manual*.

NOTE

When MBX drivers are added, some display features are disabled, such as rotate, TV-Out, and camera operation.

6.6 Image Update Application

This application requires first building an Image Update application, and then using the Image Burner.

6.6.1 Building the Image Update Application

To build an image update application, use these steps:

1. Create an **APP** folder under the following folder:
   `\Platform\3DS\Src`
2. Create an empty **dirs** file under the following folder:
   `\Platform\3DS\Src\App`
3. Copy the **ImgBurnerGUI** folder from the `\Support\App` folder to the `\Platform\3DS\Src\App` folder.
4. Select the File View tab of the Platform Builder Workspace window.
5. Expand **Favorites** > **3DS PLATFORM** > **src** > **App**.
6. Right-click on the **ImgBurnerGUI** folder and select **Clean Before Building**.
7. Right-click on the **ImgBurnerGUI** folder again and select **Build Current Project**.

   The **ImgBurnerGUI** execution file will be created in the workspace release directory.

6.6.2 Using the Image Burner

Image Burner is a simple application with only three buttons. Table 6-3 describes these controls.

Table 6-3 Image Burner Controls

Control	Description
Bin > Nb0	Converts a BinFS file (*.bin) into a binary file (*.nb0).
Low Format button	Formats the NAND Flash.
Burn nb0 button	Writes a binary image (*.nb0) into NAND Flash.