i.MX31 PDK 1.5

Windows Embedded CE 6.0

User's Guide

Document Number: 926-77200
Rev. 1.5
02/2009

ARM POWERED

freescale™ semiconductor

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

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 Freescale Semiconductor 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. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. Microsoft and Windows are registered trademarks of Microsoft Corporation.

ARM11 and ARMV41 are trademarks of ARM Limited. The ARM logo is a registered trademark of ARM Ltd.

© Freescale Semiconductor, Inc. 2007-2009. All rights reserved.

About This Book

This guide explains how to build and execute Windows Embedded CE 6.0 OS images for the 3-Stack board, using the Freescale i.MX31 PDK Windows Embedded CE 6.0 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	Describes the contents and organization of the BSP.
Chapter 2	Explains how to install/uninstall the BSP.
Chapter 3	Explains how to configure the OS images.
Chapter 4	Explains how to build the OS images and download the bootloader.
Chapter 5	Describes how to download the bootloader.
Chapter 6	Explains how to download the OS images.
Chapter 7	Describes the Windows Embedded CE 6.0 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 Hardware User's Guide
2. i.MX Advanced ToolKit Standard Version User's Guide

Chapter 1
Introduction

Using this document, you perform the following tasks in sequential order:

- Install the development tools and BSP
- Configure the OS images
- Build the OS images and bootloader
- Download the bootloader
- Download the OS images
- Use the applications

The Freescale i.MX31 3-Stack BSP is a collection of code and support files that can be integrated into the Microsoft Platform Builder development environment to create Windows Embedded CE 6.0 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

1.1 System-on-Chip Support Package (SoC)

The Freescale SoC directory contains a collection of chipset-level code that you can leverage to develop platforms based on the i.MX31 SoC. You can use the driver code and definitions in the SoC directory in a new platform design for the i.MX31 without modification.

To keep the SoC sources platform-agnostic, drivers in the SoC directory utilize hardware abstraction routines that must be ported to a specific platform or board. The SoC 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 SoC source code for the MX31:

```
WINCE600\PLATFORM\COMMON\SRC\SOC\FREESCALE\COMMON_FSL_V1
WINCE600\PLATFORM\COMMON\SRC\SOC\FREESCALE\MXARM11_FSL_V1
WINCE600\PLATFORM\COMMON\SRC\SOC\FREESCALE\MX31_FSL_V1
```

The driver code is reusable:

- The SoC driver code in the MXARM11_FSL_V1 directory is reusable across all Freescale ARM11-based SoCs.
- The SoC driver code in the MX31_FSL_V1 directory is reusable across all platforms based on the i.MX31.

1.2 Production Quality OAL (PQOAL)

Where possible, the Freescale BSP leverages the PQOAL architecture and components provided by Microsoft to reduce the OAL code that needs to be modified and maintained by the OEM. In addition, PQOAL components customized for the i.MX31 are available in the following directories:

```
WINCE600\PLATFORM\COMMON\SRC\ARM\FREESCALE\MXARM11
WINCE600\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.

1.3 i.MX31 3-Stack Platform Files

All of the driver and OAL content for the i.MX31 3-Stack hardware platform is located in the following directory:

```
WINCE600\PLATFORM\iMX313DS
```

The files in this directory implement the hardware abstraction routines invoked by the driver code in the Freescale SoC directory, as well as certain aspects of the PQOAL that may need to be modified by the OEM for a specific platform.

1.4 Sample Workspace

Platform Builder uses design solutions to encapsulate the OS components and build options needed for the Windows Embedded CE 6.0 tools to generate an OS image.

The BSP provides a default solution, which is located in the following directory:

```
\WINCE600\OSDesigns\iMX313DSMobility
```

1.5 Support Files

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

```
\WINCE600\SUPPORT
```

This directory contains applications and tests for the i.MX31 3-Stack BSP, and support files needed to configure development tools and the ATK for the i.MX31 3-Stack board.

Chapter 2
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.

This chapter explains how to install the development tools and BSP, remove an existing BSP installation, and load a sample OS design solution.

NOTE

Before installing this BSP, check to ensure that no other i.MX BSP is installed in your system. If there is an installation, see the appropriate PDK User's Guide for instructions on removing the installation.

2.1 Installing the Development Tools and BSP

This section explains how to install the Microsoft Windows Embedded CE 6.0 development tools and the Freescale i.MX31 3-Stack BSP. These installations provide a complete development environment.

To perform the installations, use these steps:

1. Install Microsoft Visual Studio® 2005, and also install the Visual Studio Service Pack 1. The Service pack may be downloaded from the following location:

   http://www.microsoft.com/products/info/product.aspx?view=22&pcid=b171ab31-2771-41f6-8243-17654838fd72&crumb=srch&qu=visual+studio+2005&gpid=f5459860-188a-46a3-a825-99fc43a9c5c1

   For installation instructions, see the *Release Notes* on the Visual Studio and Platform Builder installation discs.

2. Insert the Windows Embedded CE 6.0 installation disk.

   Optionally, you can download Windows Embedded CE 6.0 from the following link:

   http://www.microsoft.com/downloads/details.aspx?familyid=7E286847-6E06-4A0C-8CAC-CA7D4C09CB56&displaylang=en

The initial screen is displayed (Figure 1-1).

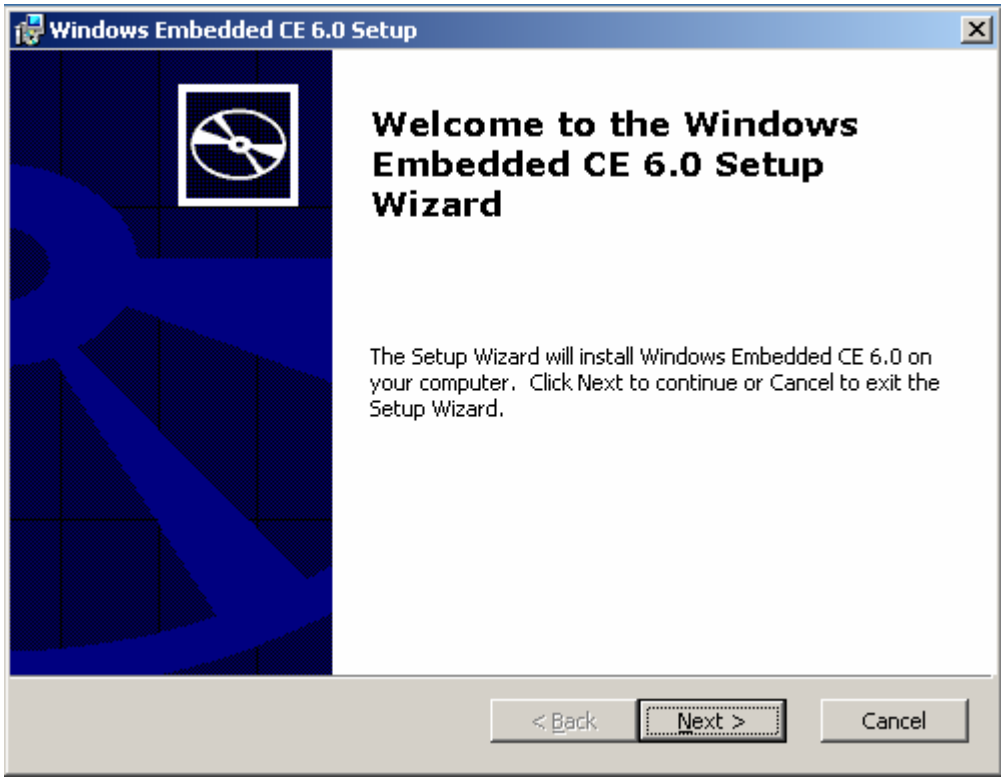

Figure 2-1 Setup Screen

3. Proceed with the installer instructions to the Setup section (Figure 1-2), and in the **CE 6.0 Operating System** folder, ensure that the ARM4I™ platform is selected.

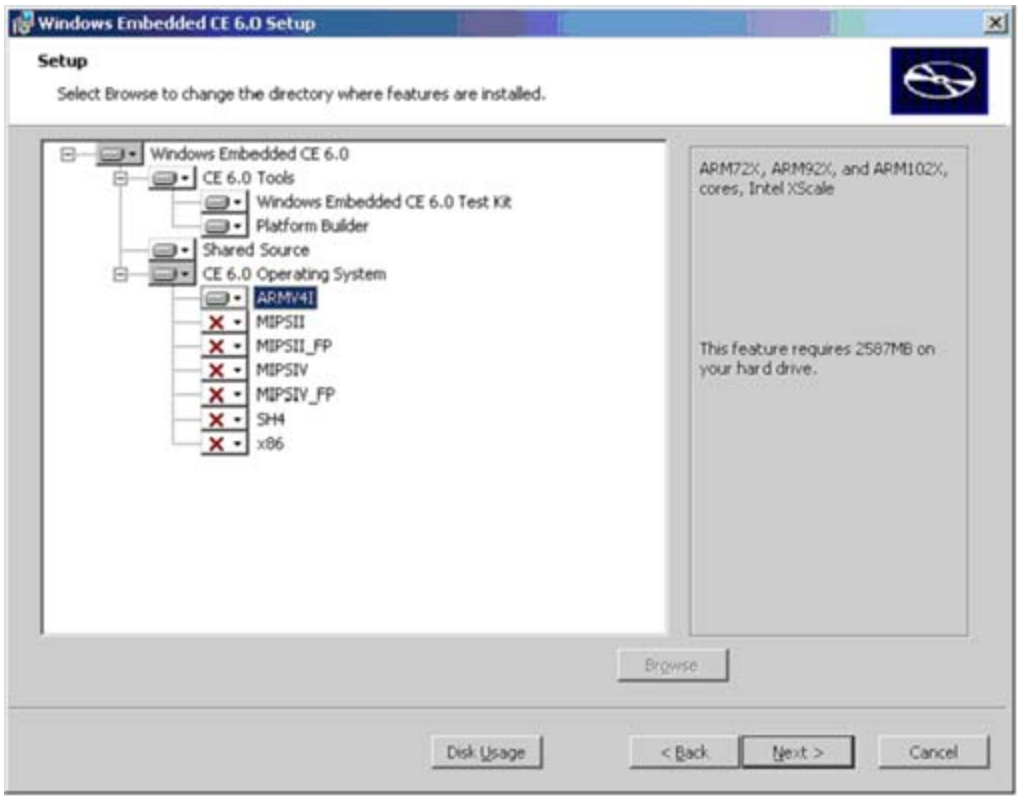

Figure 2-2 Selecting ARM41

NOTE

The ARMV4I version must be installed on the local hard drive for this BSP. No other versions are required.

4. Click **Next**.
5. Follow the installation until it is complete.
6. Install the following updates, which are required by SDK:
   Windows Embedded CE 6.0 Platform Builder Service Pack 1
   http://www.microsoft.com/downloads/details.aspx?familyid=BF0DC0E3-8575-4860-A8E3-290ADF242678&displaylang=en

Windows Embedded CE 6.0 R2
http://www.microsoft.com/downloads/details.aspx?FamilyID=f41fc7c1-f0f4-4fd6-9366-b61e0ab59565&displaylang=en

Windows Embedded CE 6.0 Cumulative Product Update Rollup (12/31/2007)
http://msdn.microsoft.com/en-us/embedded/aa731256.aspx

Windows Embedded CE Updates
Beside previous update requirements, this BSP also need and all 2008 Monthly Updates. Please install all available updates from January 2008 to November 2008.
http://msdn.microsoft.com/en-us/embedded/aa731256.aspx

NOTE

By the time you install this BSP, it is probable that a new Cumulative Product Update Rollup (2008) will be available on the Windows Embedded CE Updates site, replacing the previous Cumulative Product Update Rollup (2007). The new package includes all previous updates and would be the only file you need to install for a proper system update.

2.2 Removing i.MX31 PDK BSP

These instructions remove an installation of the BSP from the Windows Embedded CE 6.0 source code tree and the 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. Perform one of the following steps:
   — Rerun the original MSI installer and click **Remove.**

   OR

   — Using the **Add or Remove Programs** Control Panel applet, select the Freescale BSP item, and then click **Remove**.
3. Manually remove any remaining BSP files and directories (some of these files and directories will remain after completing the previous step because they contain object files or other generated files that were not part of the original BSP installation):

```
\WINCE600\OSDesigns\iMX313DSMobility
\WINCE600\PLATFORM\COMMON\SRC\ARM\dirs
\WINCE600\PLATFORM\COMMON\SRC\ARM\FREESCALE
\WINCE600\PLATFORM\COMMON\SRC\SOC\FREESCALE
\WINCE600\PLATFORM\iMX313DS
\WINCE600\SUPPORT
```

4. Manually remove the residual BSP library files that were created after the BSP was installed. To find the libraries, use the Windows Explorer search functionality. Search for `*mxarm11*; *mx31*; *pmic*; *mc13783*` in the following library path, and then remove all the `LIB`, `PDB`, and `DEF` files found by the search:

   ```
   \WINCE600\PLATFORM\COMMON\LIB\ARMV4I
   ```

5. Restore the original version of the following file:
   `WINCE600\PLATFORM\COMMON\SRC\ARM\dirs`

2.3 Load Sample OS Design Solution

After installing the BSP, you can use the sample OS solutions provided in the BSP package to build a Windows Embedded CE 6.0 OS image.

To load the OS image, use these steps:

1. In Platform Builder, select **File** > **Open** > **Project/Solution**…
2. Select the i.MX31 BSP sample workspace by loading the following file:

   ```
   WINCE600\OSDesigns\iMX313DSMobility\iMX313DSMobility.sln
   ```

   The process of loading this solution also loads the associated i.MX31 BSP catalog.
3. Select the Catalog Items View tab to see all of the available OS solution catalog items.

Chapter 3
Configuring OS Images

This chapter describes the configuration options available for the i.MX31 BSP. You will use Platform Builder to select one of the two default build configurations provided with the BSP sample solution. These configurations control the type of OS image (debug or release) that will be generated by the Platform Builder tools.

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

Use Platform Builder to select a configuration.

NOTE

The sample solution provided in the BSP is correctly configured to generate a default image targeted for the i.MX31 3-Stack board. 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 select the type of OS image to be generated by Platform Builder, use these steps:

1. Click **Build > Configuration Manager.**
2. Select **Set Active Configuration**. The sample workspace provided with the BSP provides two image build types:

   Freescale i.MX31 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 i.MX31 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 Embedded CE 6.0, see the "Build Configurations" topic in the Windows Embedded CE 6.0 Help.

3.2 Configuring the BSP Environment Variables

The BSP environment variables control the optional BSP support for OS images. Select the BSP environment variables using the **Project** > **Properties** > **Configuration Properties** > **Environment** configuration dialog and the i.MX31 3-Stack BSP Catalog.

The subsections that follow explain how to configure the BSP environment variables and describe the supported variables.

3.3 Configuration Properties Environment Variables

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

To view and configure the variables, use these steps:

1. Open the sample solution for the BSP.
2. From the Project menu, select **iMX313DSMobility Properties**.
3. Expand **Configuration Properties** if necessary and select **Environment**.

Note

After modifying the environment value in the Properties window, also change the related value in the following location:

`\wince600\platform\iMX313DS\imx313ds.bat`

For a Release build type, all environment values are defined in this `bat` file.

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

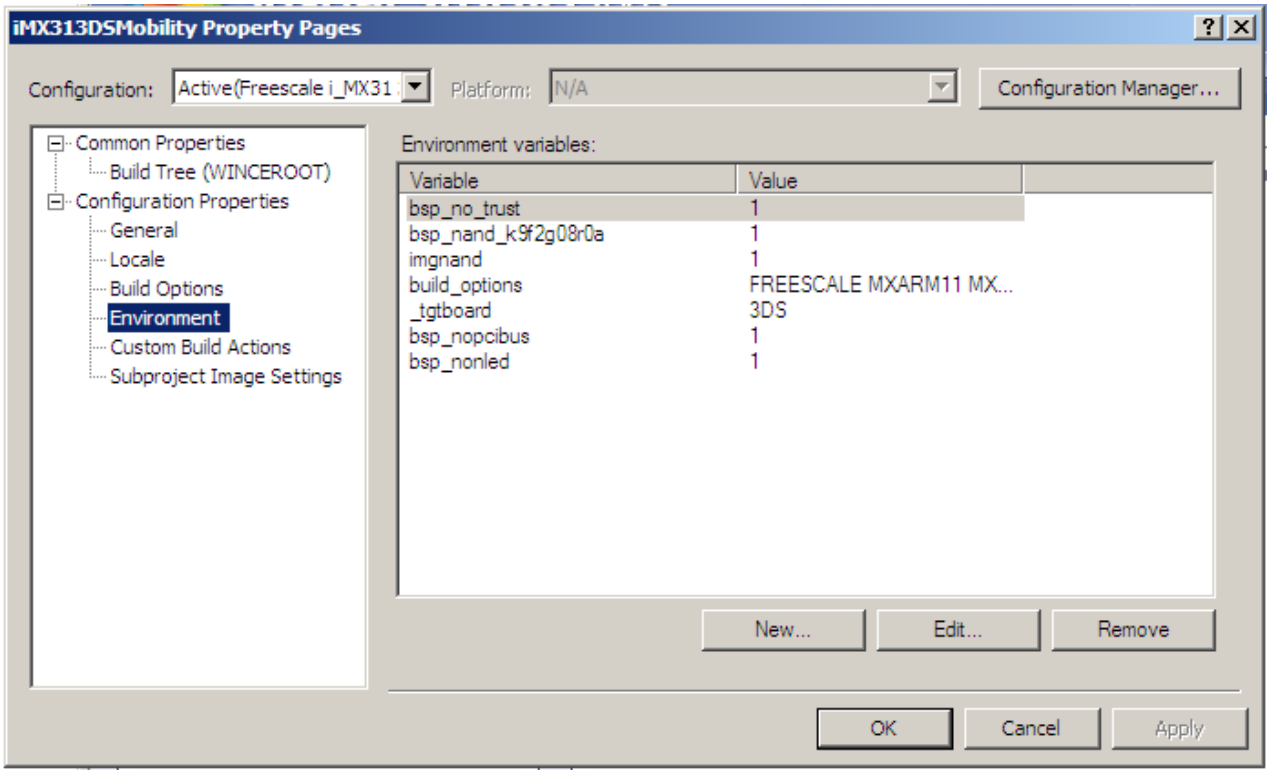

Figure 3-1 Environment Variables

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

BSP Variables	Description	Settings
		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.
BSP_WARNLEVEL	Selects the compiler warning/verbosity level. See the online help for the /w compiler option for additional information. Note that WARNISERROR=1 is also defined by default in the PUBLIC\COMMON\sources.cmn, PLATFORM\COMMON\sources.cmn, and the PLATFORM\iMX313DS\sources.cmn files so that any compiler warnings will be treated as build errors.	BSP_WARNLEVEL=4 This means that compiler level 4 warnings will be displayed and treated as build errors if WARNISERROR=1 is also defined.
_TGTBOARD	Allows conditional compilation for the i.MX31 3-Stack hardware.	_TGTBOARD = 3DS 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 MXARM11_FSL_V1 MX31_FSL_V1 PMIC Required for proper build of BSP. DO NOT modify or remove from sample workspace.
BSP_NAND_K9F2G08R0A	Used by EBOOT and OS build files	BSP_NAND_K9F2G08R0A = 1 DO NOT 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 Variables	Description	Settings
BSP_HIVE_ATA	Used to build a HIVE image that uses ATA as storage media	BSP_HIVE_ATA = 1 Note: SYSGEN_FSREGHIVE and SYSGEN_FSROMONLY should be added in the solution to build a HIVE image, need clean build to generate image
BSP_HIVE_SDHC	Used to build a HIVE image that uses SD as storage media	BSP_HIVE_SDHC = 1 Note: SYSGEN_FSREGHIVE and SYSGEN_FSROMONLY should be added in the solution to build a HIVE image, , need clean build to generate image
BSP_HIVE_NAND	Used to build a HIVE image that uses NAND as storage media	BSP_HIVE_NAND = 1 Note: SYSGEN_FSREGHIVE and SYSGEN_FSROMONLY should be added in the solution to build a HIVE image, need clean build to generate image
BSP_USB_IRAM_PATCH	Used to build a USB image that use IARM instead RAM	BSP_USB_IRAM_PATCH = 1

Chapter 4
Building OS Images and Bootloader

This chapter explains how to build Windows Embedded CE 6.0 OS images using the i.MX31 3-Stack BSP.

4.1 Building SoC Libraries

The Freescale SoC libraries that support the i.MX31 are generated during the following build procedure:

Build > **Advanced Build Commands** > **Sysgen** or **Build** > **Advanced Build Commands** > **Build Current BSP and Subprojects**

Windows Embedded CE 6.0 ships with pre-built SoC libraries for various ARM processors, but the libraries for i.MX31 must be built from the sources, because they are not included with the standard Microsoft distribution.

Note

The provided sample OS design solution is preconfigured to build the required Freescale SoC. That is, the sample i.MX31 OS design solution will automatically build the i.MX31 SoC sources.

To build the SoC libraries, use these steps:

1. Open the sample solution.
2. Select the build type, as discussed in Section 3.1, Selecting the Image Build Type.
3. If you have not yet performed a **Sysgen** operation, select a **Sysgen** build.

   If you have already performed a **Sysgen** operation, and just need to rebuild the Freescale SoC libraries, click **Build Current BSP and Subprojects**.

   For a release build type, the SoC libraries will be placed in:
   `WINCE600\PLATFORM\COMMON\LIB\ARMV4I\RETAIL`

   For a debug build type, the SoC libraries will be placed in:
   `WINCE600\PLATFORM\COMMON\LIB\ARMV4I\DEBUG`

4.2 Building Run-Time Images

For instructions for the remaining steps of building an OS image, follow the standard procedures described in the Platform Builder documentation. For information, see the "Building a Run-Time Image" topic in the Windows Embedded CE 6.0 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 solution. and select the desired build type, as discussed in Section 3.1, Selecting the Image Build Type.
2. Using the **Build** > **Global Build Settings** menu, configure the build options as follows:
   - **Copy Files to Release Directory After Build** selected
   - **Make Run-Time Image After Build** selected
3. Click **Build** > **Advanced Build Commands** > **Sysgen** to start the build (Figure 4-1).

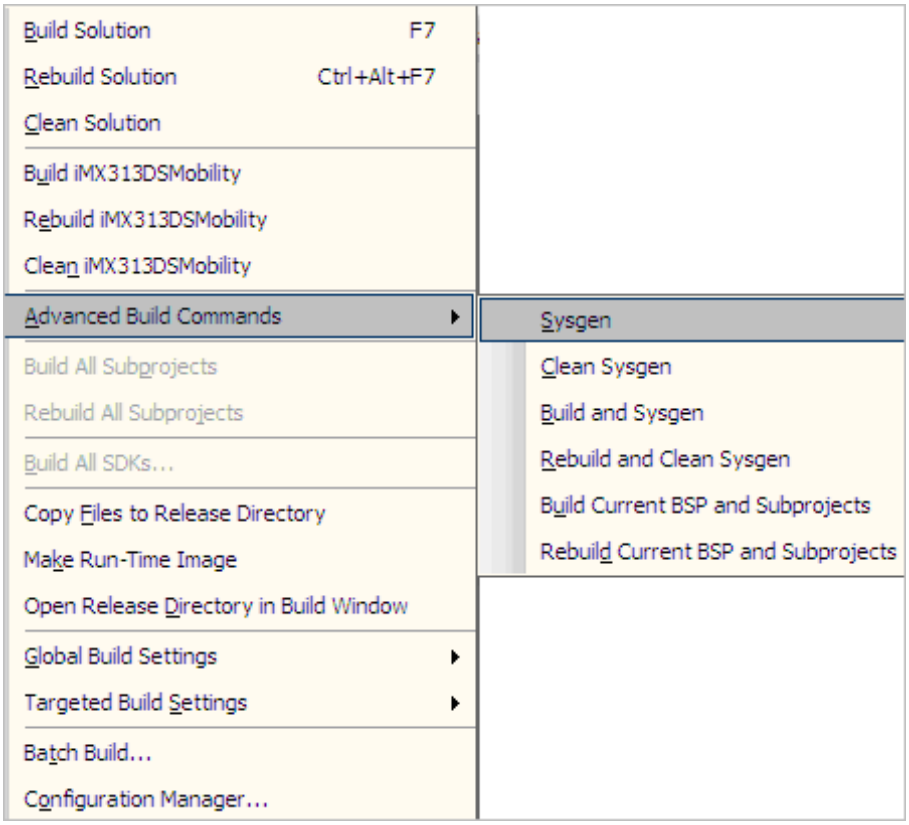

Figure 4-1 Building the BSP for the First Time

For a release build type, the resulting OS image files are placed in the following directory:
`WINCE600\OSDesigns\iMX313DSMobility\RelDir\Freescale_i_MX31_3DS_ARMV4I_Release`

For a debug build type, the resulting OS image files are placed in the following directory:
`WINCE600\OSDesigns\iMX313DSMobility\RelDir\Freescale_i_MX31_3DS_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. However, as some time you may want to force a clean build for the BSP.

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

1. Using the Build OS menu, configure the build options as follows.
   - **Copy Files to Release Directory After Build** selected
   - **Make Run-Time Image After Build** unselected

2. Click **Build** > **Advanced Build Commands** > **Rebuild Current BSP and Subprojects** to perform a clean build of the BSP platform directory (including the SoC libraries) and complete the creation of a new OS image. (Figure 4-2).

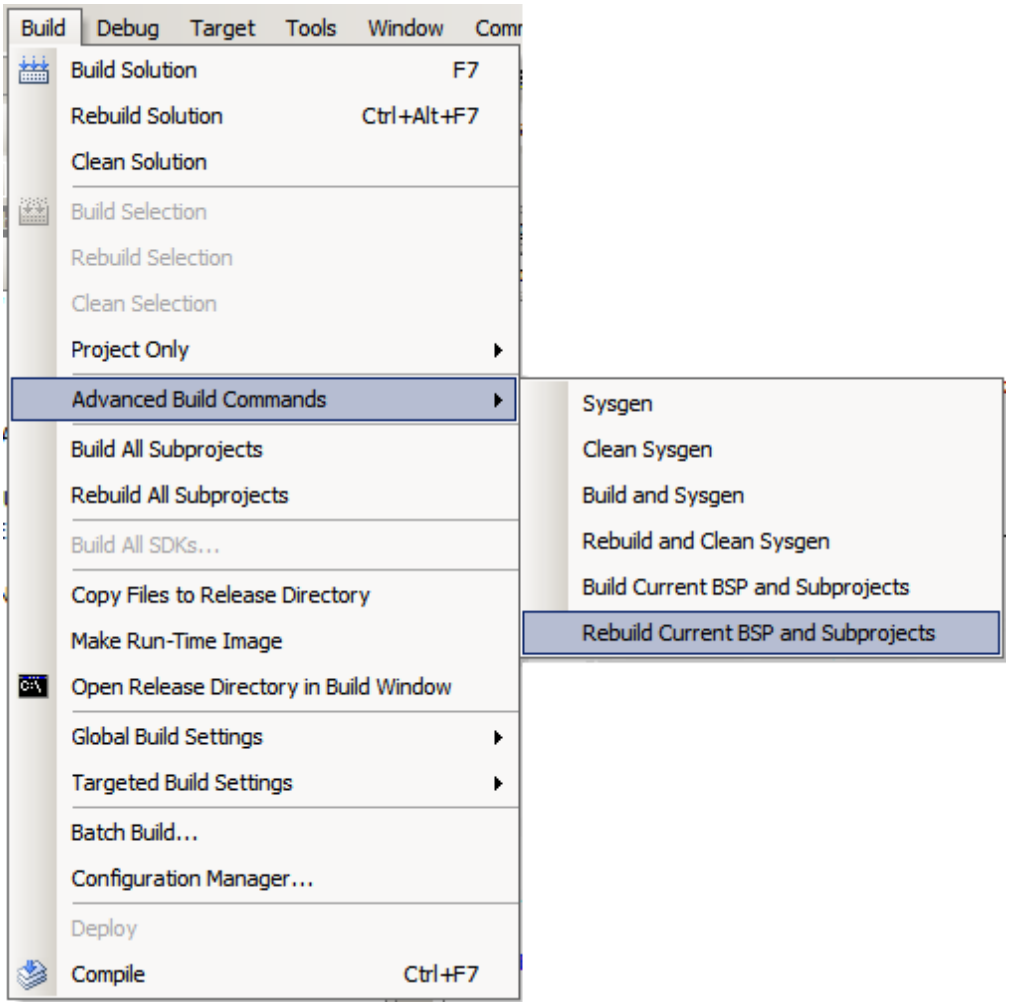

Figure 4-2 Clean Build for the BSP

4.2.3 Performing an Incremental BSP Build

The **Sysgen** build phase copies the pre-built OS component binaries 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 solution.
2. Select the desired build type, as discussed in Section 3.1, Selecting the Image Build Type.
3. Using the **Build** > **Global Build Settings** menu, configure the build options as follows:
   - **Copy Files to Release Directory After Build** selected
   - **Make Run-Time Image After Build** selected

4. Click **Build** > **Advanced Build Commands** > **Build Current BSP and Subprojects** to perform an incremental build of the BSP platform directory (including the SoC libraries) and complete the creation of an OS image (Figure 4-3).

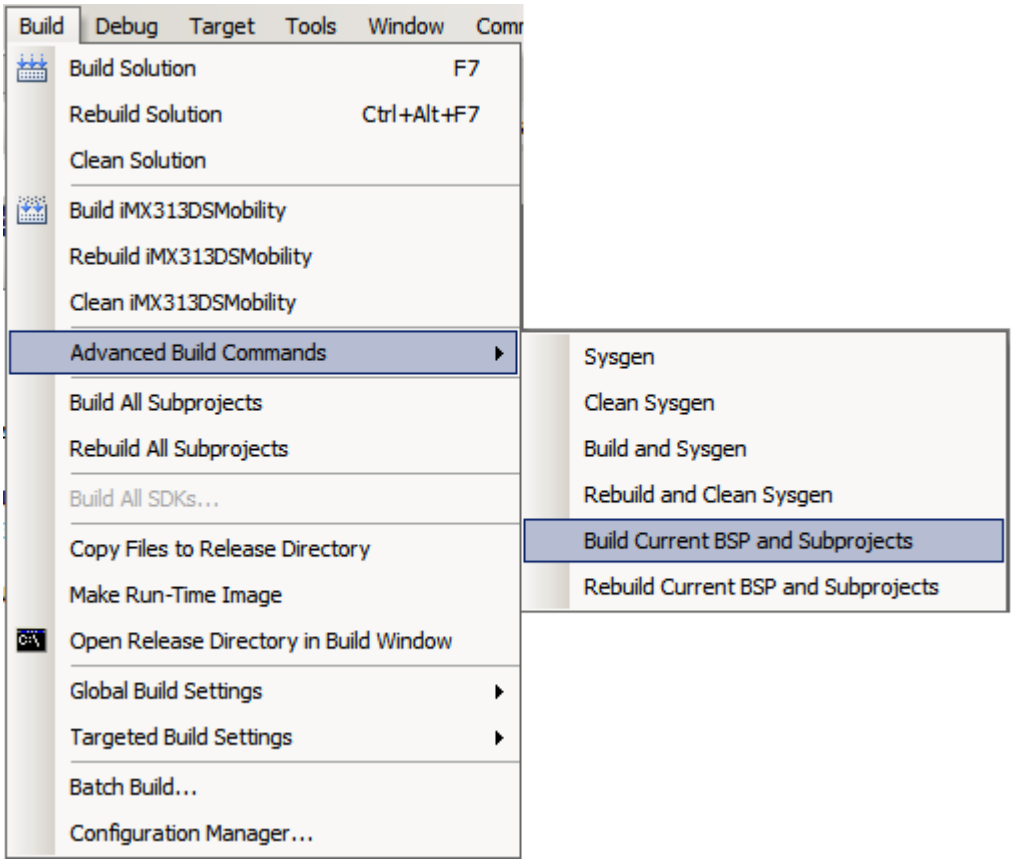

Figure 4-3 Incremental BSP Build Libraries

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), which provides additional functionality 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. Using the **Build** menu, configure the options as follows:
   - **Copy Files to Release Directory After Build** selected
   - **Make Run-Time Image After Build** unselected
2. Select the Solution Explorer tab of the Platform Builder Solution Explorer window.
3. Expand **WINCE600** > **PLATFORM** > **iMX313DS** > **src** > **BOOTLOADER**.

4. Right-click on the XLDR folder and select **Rebuild**.

   The XLDR binary image is created in the solution release directory (Figure 4-4).

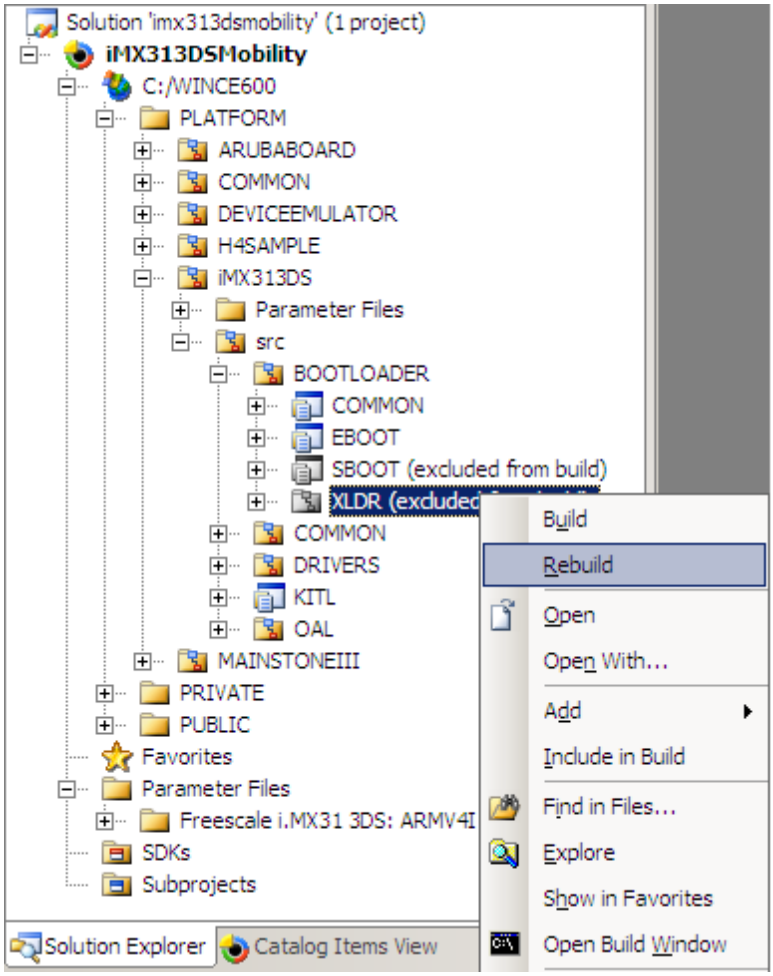

Figure 4-4 Building an XLDR Image

Chapter 5
Downloading the Bootloader

To prepare for downloading the OS images and debugging, you configure and initialize the target and development workstations.

These procedures include configuring boot mode on the Debug board, and then programming XLDR and EBOOT.

5.1 Configuring Boot Mode

To configure boot mode, set the switches on the Debug board as described.

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 internal ROM boot configuration.

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

5.2 Programming XLDR and EBOOT Using the ATK Tool

If you receive an i.MX31 3-Stack board without XLDR and EBOOT, you must program them by using the Advanced Toolkit (ATK). Perform this task after your OS image has been built.

NOTE

Before building XLDR and EBOOT, set the environment variable IMGNAND = 1 in the following location in Platform Builder:

Project > **Properties** > **Configuration Properties** > **Environment**

You may obtain the Advanced ToolKit (ATK) from the i.MX31 BSP package. The ATK supports K9F2G08R0A.

Note

If this is your first use of the i.MX31 3-Stack board, we strongly recommend that you erase the NAND first.

To erase NAND, use these steps:

1. Set the boot mode to internal ROM boot.
2. Power on or reset the device board.
3. Run `ADSToolkit.exe` (Figure 5-1).

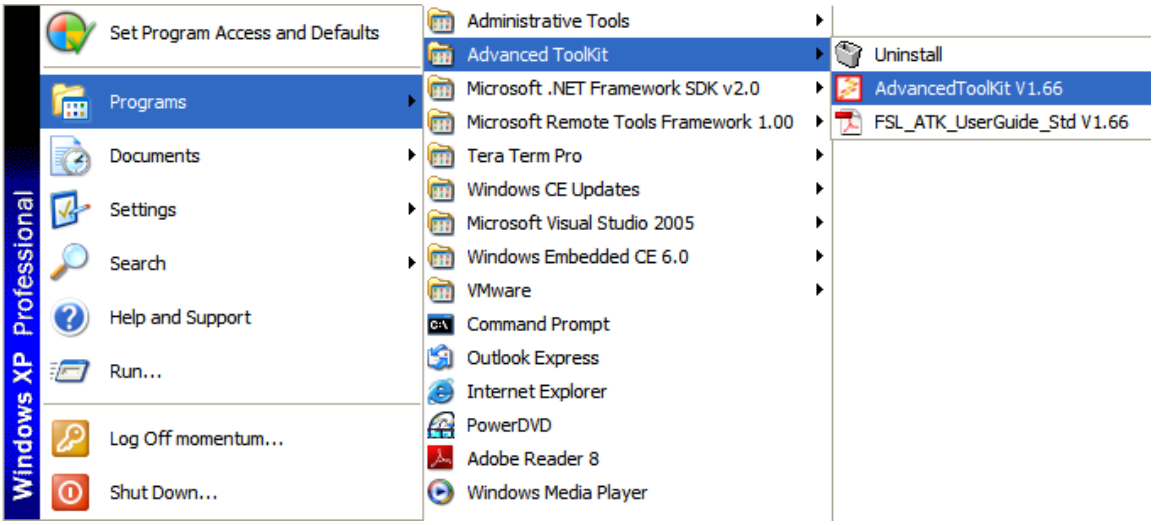

Figure 5-1 Location of Advanced Toolkit Application

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

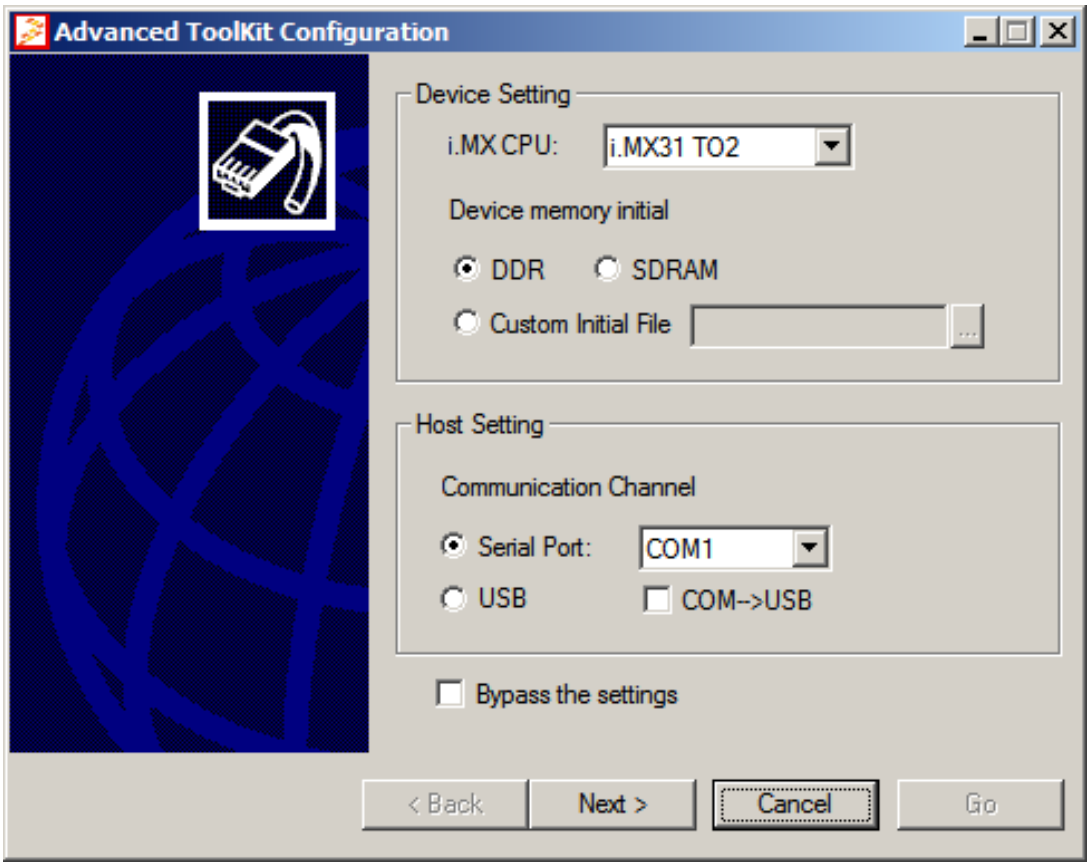

Figure 5-2 ATK Initial Configuration

4. 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**.
6. Click **Flash Tool**, and then click **Go**.

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

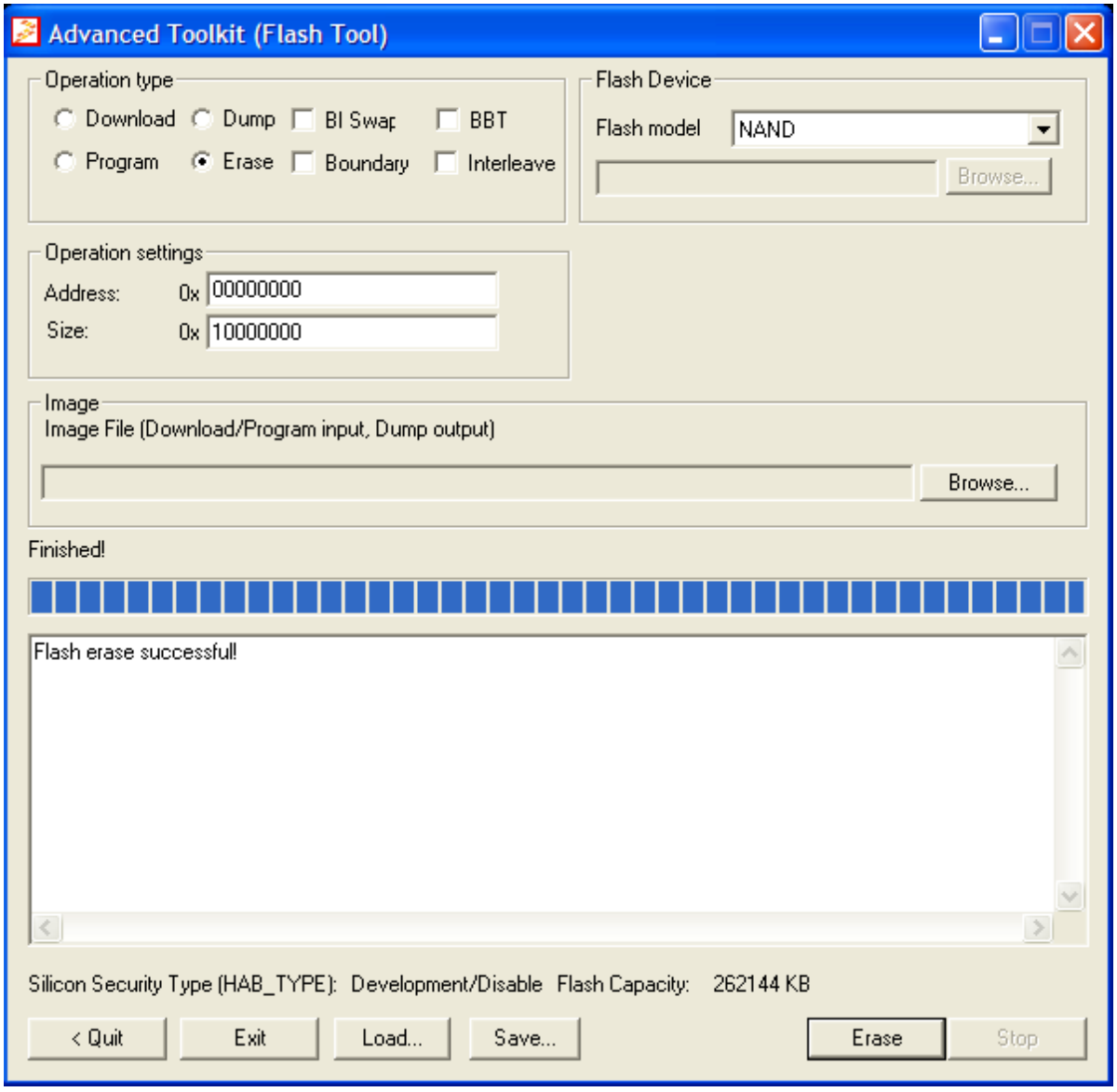

Figure 5-3 Erase Flash with ATK

5.2.1 Program XLDR

To program the XLDR, use these steps:

1. Set the boot mode to internal ROM boot.
2. Power on or reset the device board.
3. Run `ADSToolkit.exe`. The Advanced ToolKit Configuration screen is displayed (Figure 5-4).

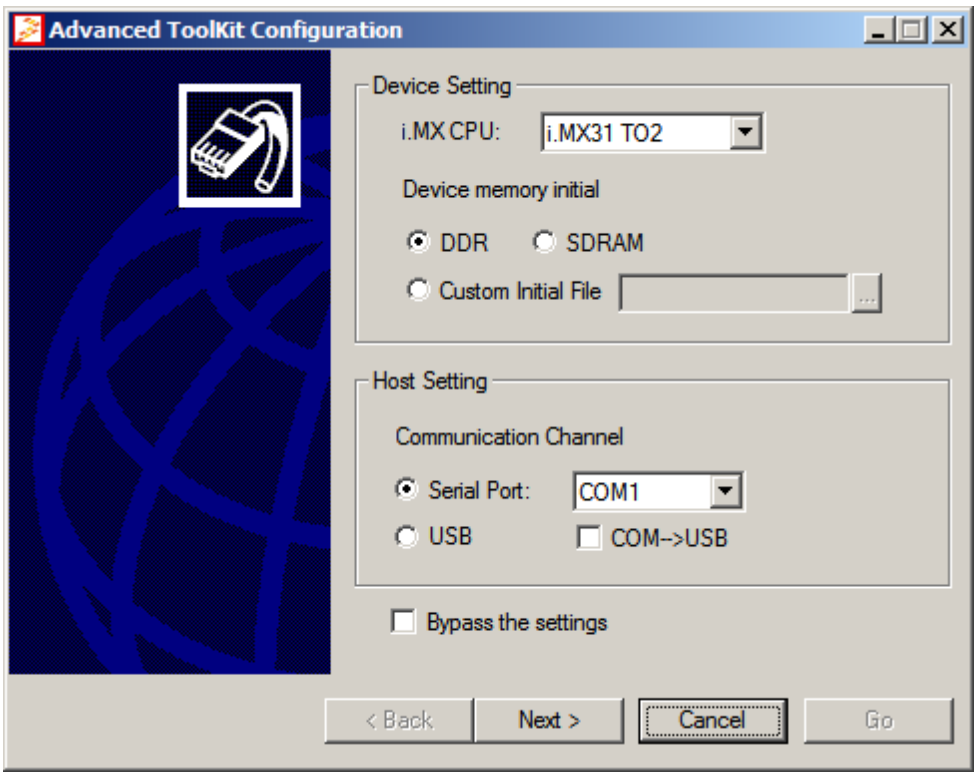

Figure 5-4 Initial Configuration

4. Select the following:
   - 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.
   - **Bypass the settings** can be left blank.
5. Click **Next**.
6. Click **Flash Tool**, and then click **Go**.

   The Flash Tool screen is displayed.
   - For **Operation Type**, select **Program**.
   - For **Flash model**, select **NAND**.
   - Configure **Address** as **0x0**.
   - For **Image File**, click **Browse** to select `XLDR.nb0`.

Click **Program**, and wait until flashing is successful, as shown in Figure 5-5.

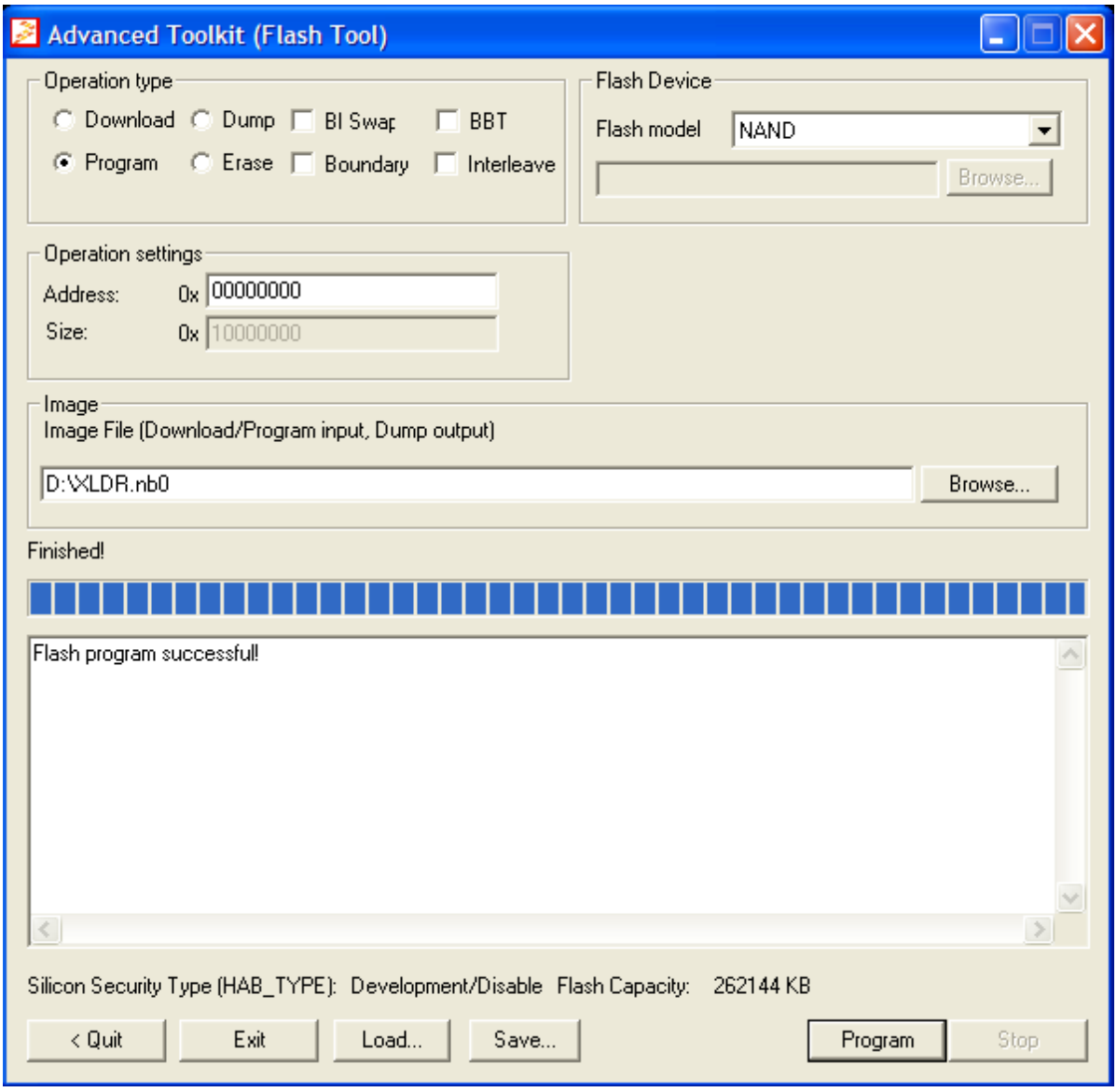

Figure 5-5 XLDR Programming

5.2.2 Program EBOOT

When the ATK finishes programming **XLDR.nb0** to the board, program **EBOOT.nb0** to the board.

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

1. Power on or reset the device board.
   - For **Operation type**, select **Program** and **Bl Swap**.
   - For **Flash model**, select **NAND**.
   - For **Operations** settings, set the address as **0x20000**.
   - For **Image File**, click **Browse** to select **EBOOT.nb0**.
2. Click **Program** and wait until flashing is successful, as seen in Figure 5-6.

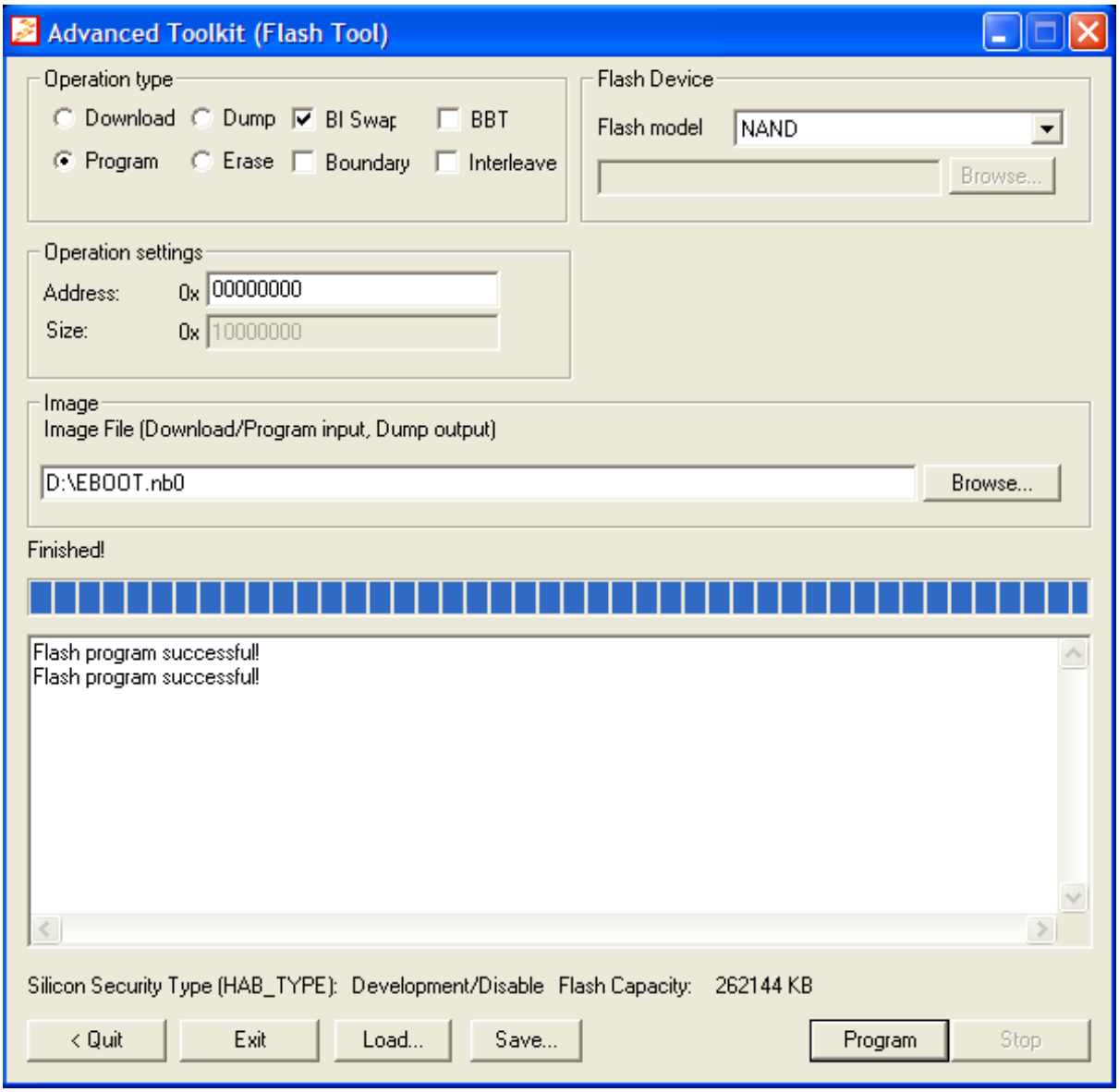

Figure 5-6 XLDR.nb0 Programming

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.

5.3 Updating XLDR and EBOOT

You may use either the ATK or Platform Builder to update XLDR and EBOOT. To do so, repeat the steps in Section 5.2.1.

5.4 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 KITL passive mode, this option bypasses the KITL connection if it is enabled.
C	L2 Cache Enable	Enables/Disables the L2 cache from the ARM processor.

W	L2 Cache Mode	Configures the L2 cache mode to be WriteThrough 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.
X	Power Policy	Enable/Disable Power Policy. This option should be enabled only if you want to do a power consumption test.
R	USB KITL	Enable/Disable USB KITL when KITL Passive mode is Disabled
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	Downloads the OS Image or EBOOT image using USB RNDIS service.
L	Launch Existing Flash Resident Image Now	Launch the OS image stored in NAND.

Chapter 6
Downloading the OS Images

Now that you have added the i.MX31 3-Stack board to your Platform Builder devices, and you have downloaded the bootloader, you can download the OS image using EBOOT. Make sure that the 3-Stack board is configured for NAND Flash boot mode, as described in section 5.1.

6.1 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

6.2 Selecting the i.MX31 PDK Target Device

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 steps in this section 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 **Project** > **Properties** > **Configuration Properties** > **Environment**.

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 (Figure 6-1). If the EBOOT startup sequence is not displayed, program XLDR and EBOOT using the procedure in Section 5.2, Programming XLDR and EBOOT Using the ATK Tool.

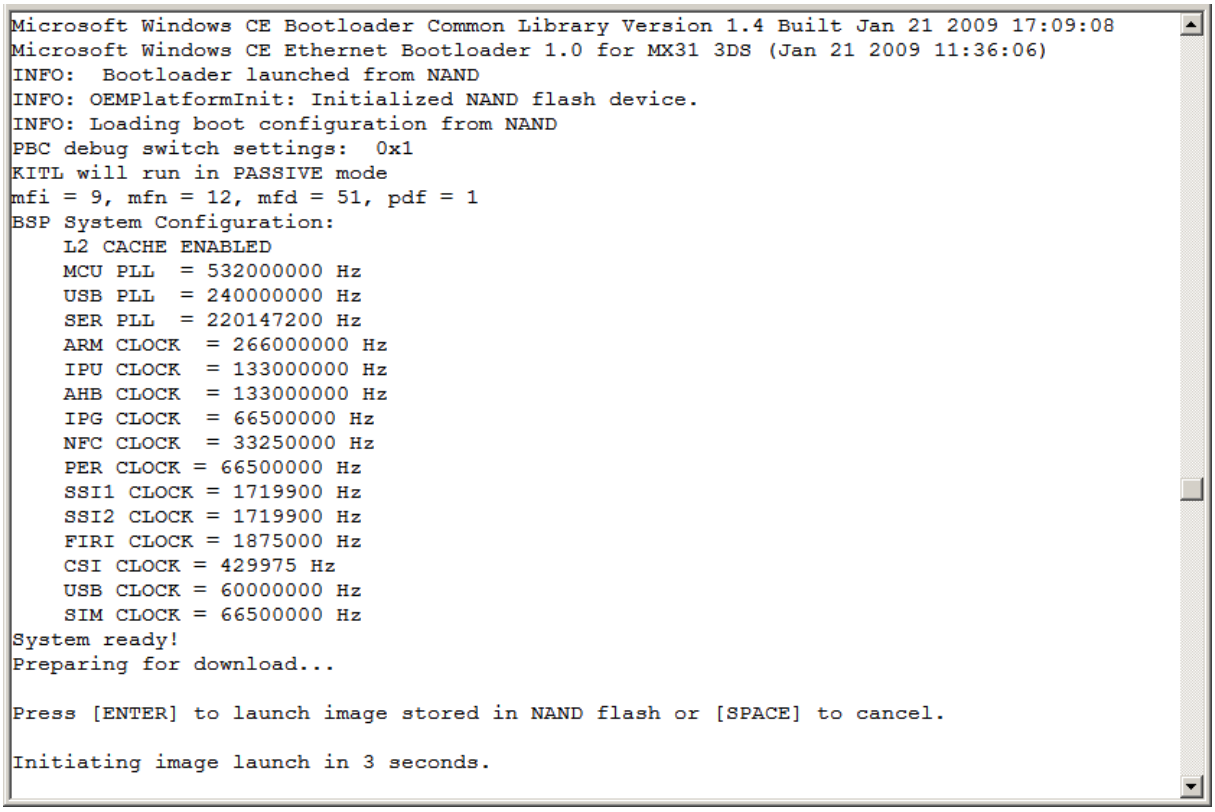

Figure 6-1 EBOOT Startup Sequence

Perform ONE of these steps:

— If you have just erased the NAND and then programmed XLDR/EBOOT, the EBOOT menu is displayed. Update the MAC Address, and then press **X** to disable Power Policy. (Enable the Power Policy only if you want to perform the power consumption test.)

OR

— Press the spacebar on the host PC to enter the EBOOT menu before the three-second period ends.

6. Configure the ETH settings to match your network configuration, for example:

   0) IP address: 192.168.0.2

   1) Subnet Mask: 255.255.255.0

   2) Boot delay: 3 seconds

   3) DHCP: Disabled

   5) Auto boot: NK from NAND

   6) MAC address: EE-04-9F-00-8C-9C

   I) KITL interrupt mode: Disable

   P) KITL passive mode: Disable

   R) USB KITL: Enable

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

7. In Platform Builder, select **Target** > **Connectivity Options** (Figure 6-2).

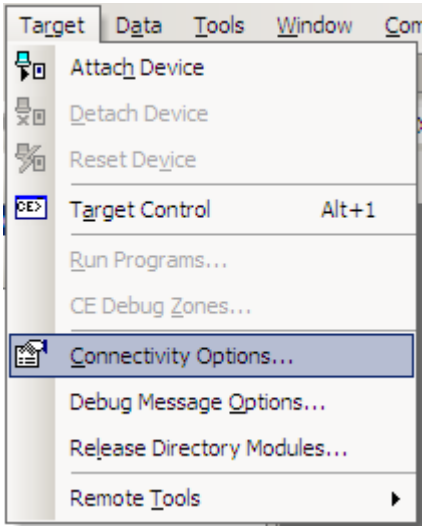

Figure 6-2 Platform Builder Connectivity Options

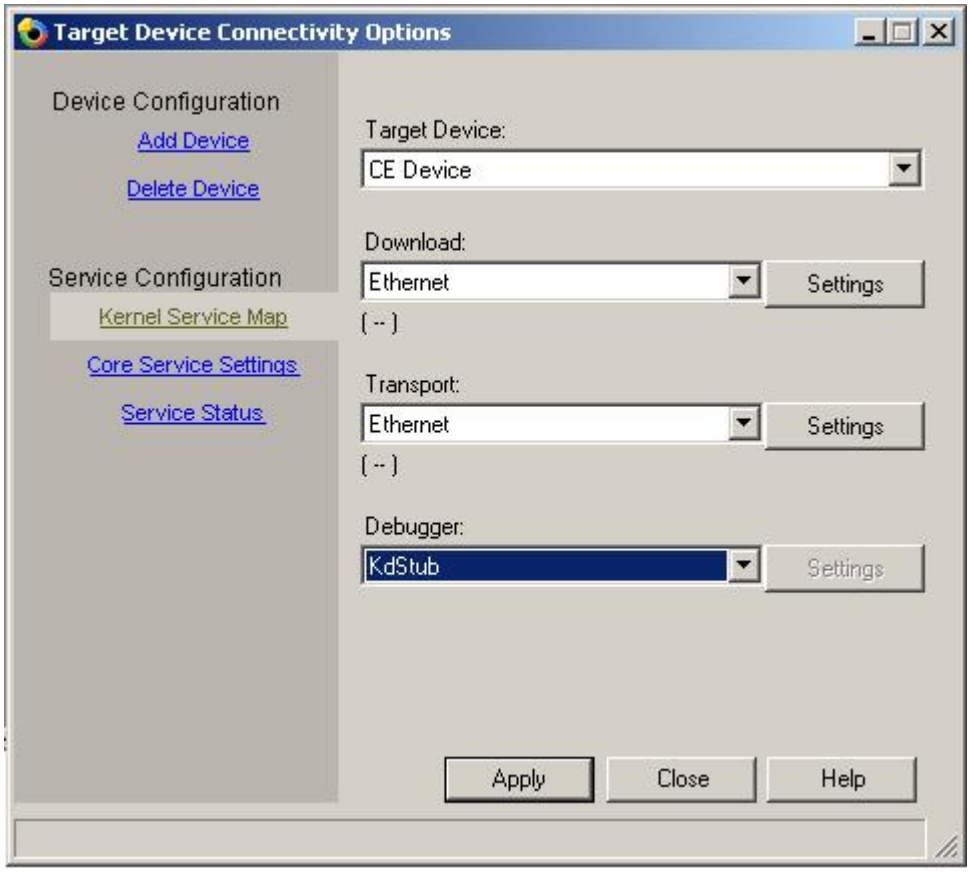

Figure 6-3 Target Connectivity Options

The Target Connectivity dialog is displayed (Figure 6-3).

8. **Select these options:**
   - For **Download** and **Transport**, select **Ethernet**
   - For **Debugger**, select **KdStub**

   The debugger should be **None** when the Kernel Debugger is not enabled in **Project** > **Properties** > **Configuration Properties** > **Build Options**.

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

The download begins, as shown in Figure 6-4.

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**.

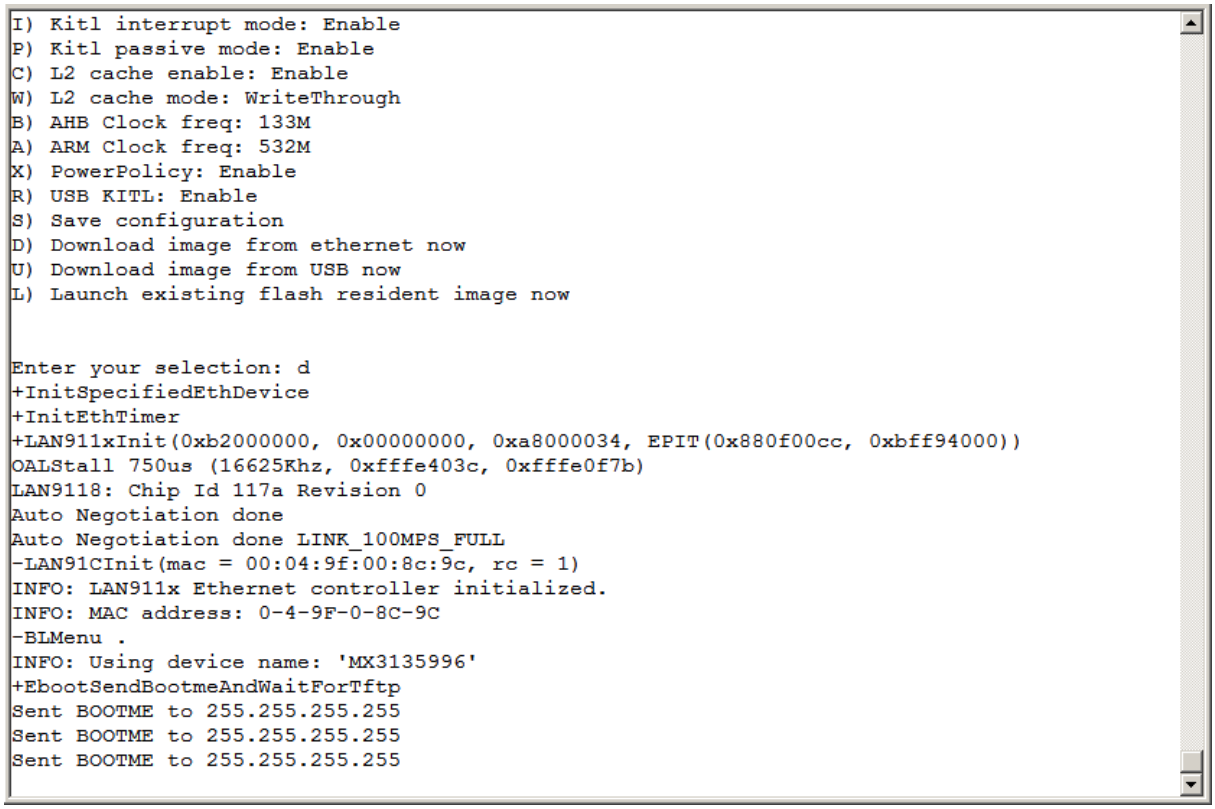

Figure 6-4 EBOOT Request Download Message Screen

10. In the Target Connectivity dialog, in the **Download** section, click **Settings**.

    The Ethernet Download Settings window is displayed.

11. When the EBOOT program displays the BOOTME messages, the 3-Stack board device should be displayed in the Ethernet Download Settings dialog (Figure 6-5). 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 BlackICE software) on the host PC.

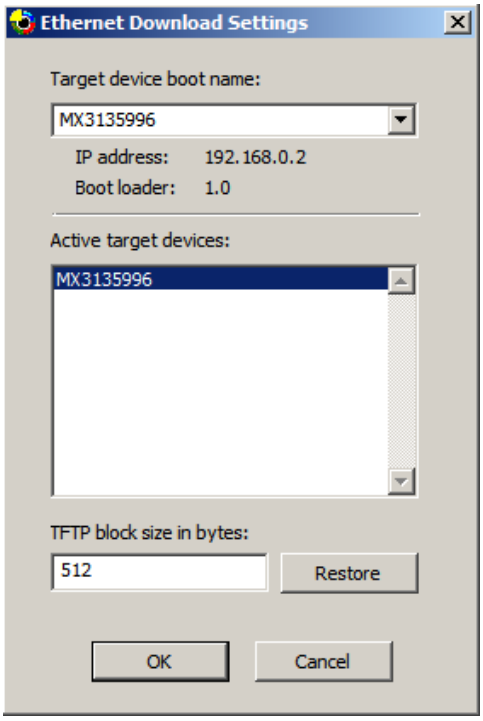

Figure 6-5 Ethernet Download Settings

After selecting a target, the Target Device Connectivity Options screen is displayed with the i.MX31 device listed under the Download box. The **Target Device** box should display **CE Device** (Figure 6-6).

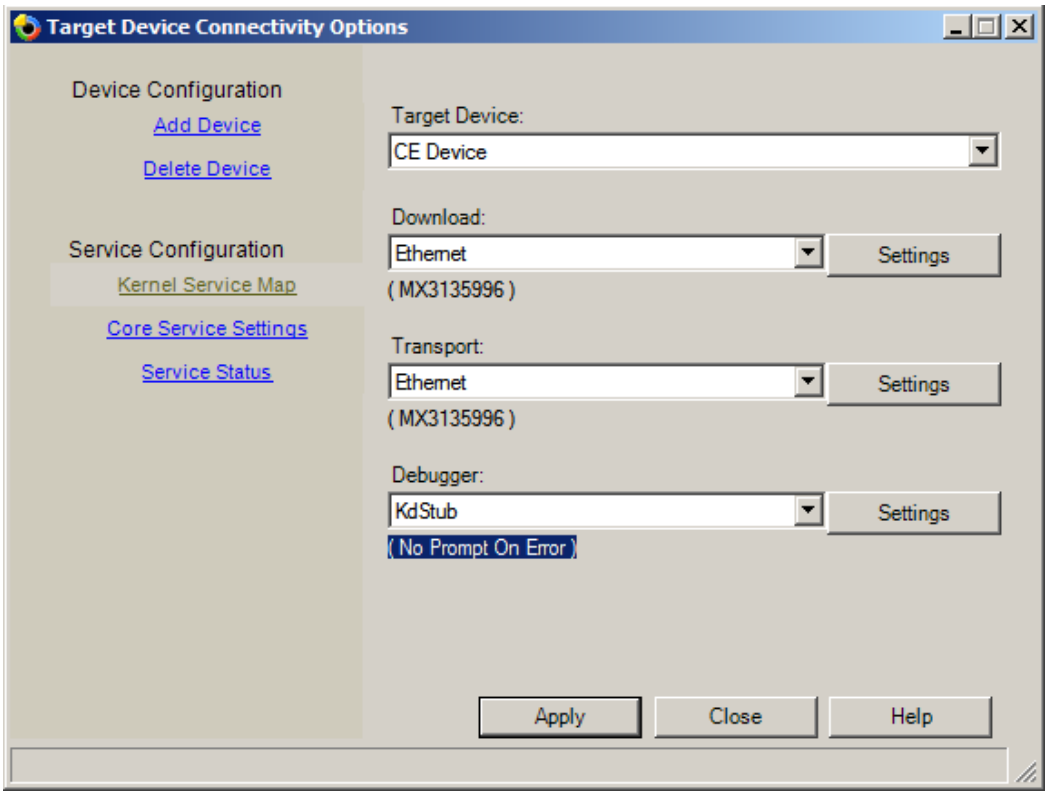

Figure 6-6 Target Device Connectivity Settings

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

6.3 Downloading the OS Kernel Image using EBOOT

If the device is already sending BOOTME signals and the i.MX31 target device is selected (section 6.2), download the OS kernel image using EBOOT.

1. Click **Target** > **Attach Device** (Figure 6-7).

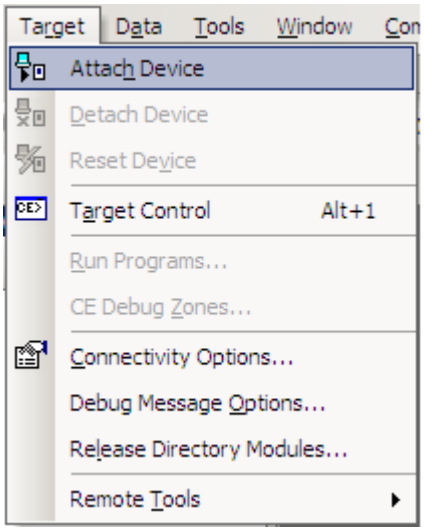

Figure 6-7 Attach Device to Download OS Image

The Download Runtime Image to CE Device dialog is displayed, showing the progress of the download (Figure 6-8).

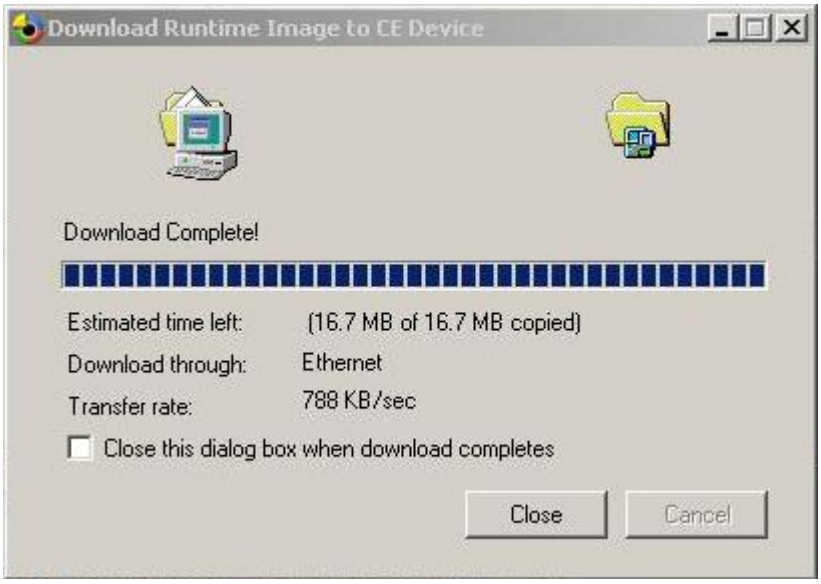

Figure 6-8 Download Complete

2. Wait until the Platform Builder finishes the OS Image download.
3. At the serial console, press the '**Y**' key to begin programming the flash (Figure 6-9).

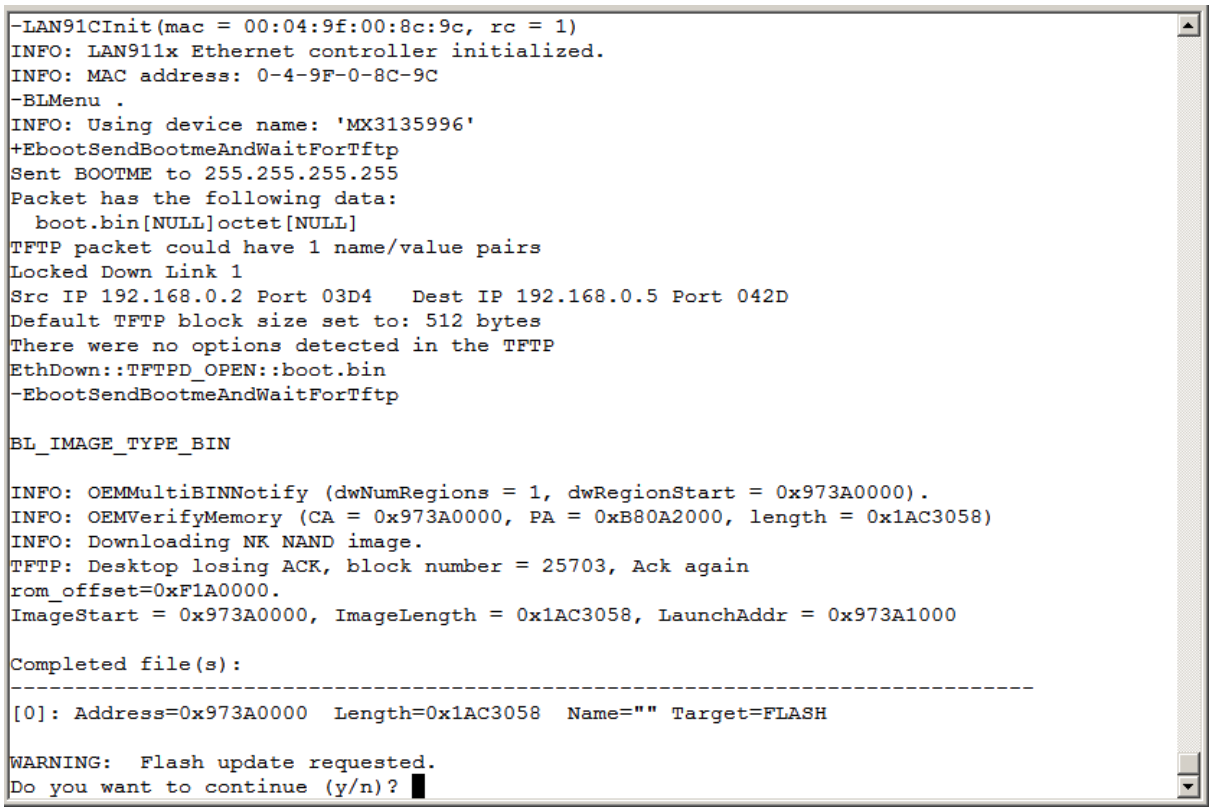

Figure 6-9 EBOOT Flash Update

EBOOT will program the image into flash and display the status using serial debug message (Figure 6-10).

```
Locked Down Link 1
Src IP 192.168.0.2 Port 03D4   Dest IP 192.168.0.5 Port 0425
Default TFTP block size set to: 512 bytes
There were no options detected in the TFTP
EthDown::TFTPD_OPEN::boot.bin
-EbootSendBootmeAndWaitForTftp

BL_IMAGE_TYPE_BIN

INFO: OEMMultiBINNotify (dwNumRegions = 1, dwRegionStart = 0x973A0000).
INFO: OEMVerifyMemory (CA = 0x973A0000, PA = 0xB80A2000, length = 0x1AC3058)
INFO: Downloading NK NAND image.
TFTP: Desktop losing ACK, block number = 25703, Ack again
rom_offset=0xF1A0000.
ImageStart = 0x973A0000, ImageLength = 0x1AC3058, LaunchAddr = 0x973A1000

Completed file(s):
-------------------------------------------------------------------------------
[0]: Address=0x973A0000  Length=0x1AC3058  Name="" Target=FLASH

WARNING:  Flash update requested.
Do you want to continue (y/n)?
INFO: Writing NK image to NAND (please wait)...
INFO: Erasing NAND flash blocks [0x5 - 0x1E5].
INFO: Programming NK image from flash cache address 0xA8100000, size = 28061784
INFO: Program is 100% complete.
INFO: Verifying image.
INFO: Read is 100% complete.
INFO: Update of NK completed successfully.
Reboot the device manually...
SpinForever...
```

Figure 6-10 EBOOT Flash Update

4. When the programming is complete, reboot the board.
5. Enter the EBOOT menu. Now the Windows Embedded CE 6.0 system can automatically boot from NAND Flash. Press **L** to launch the OS image, or reboot the board to automatically launch the OS image from NAND.

6.4 Using UBOOT

To use UBOOT, follow these steps:

1. If you want use the USB KTIL before building the kernel, first 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.

6. Press the ‘u’ key to download the image from the USB.

   If you are using UBOOT or USB KITL for the first time, you are prompted for installation options. This installs the RNDIS driver (Figure 6-11).

Figure 6-11 Installing RNDIS Driver on Windows XP

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

8. Click **Next**.

   You are prompted to select a driver for RNDIS (Figure 6-12).

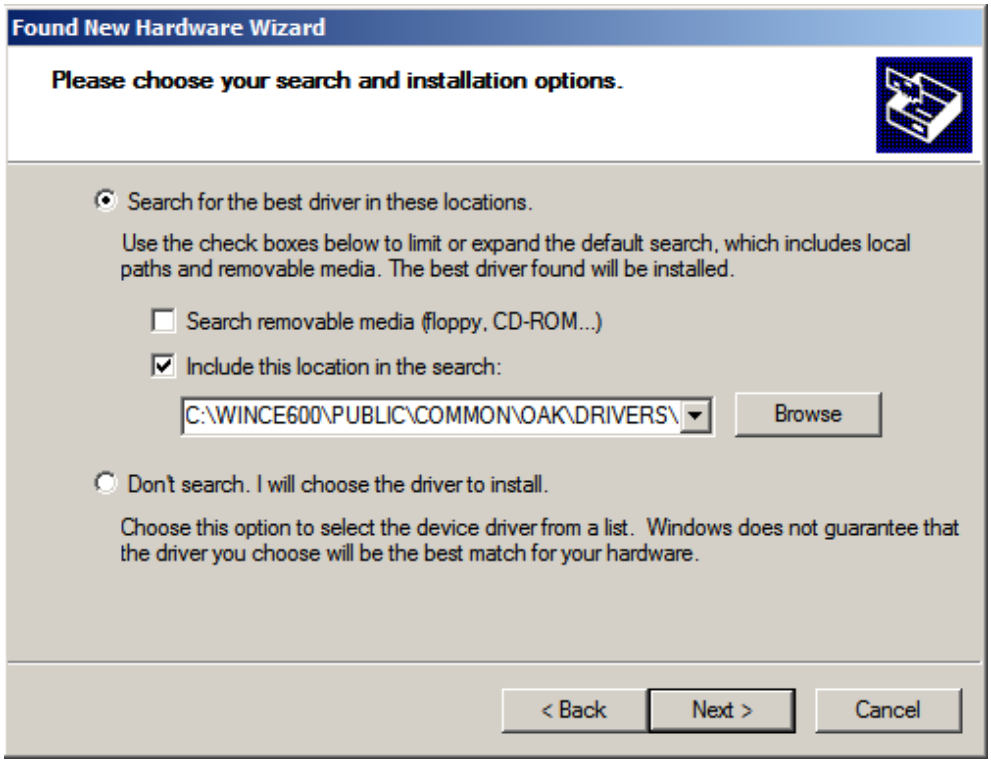

Figure 6-12 Selecting a Driver for RNDIS

9. Select **Include this location in the search** and add the following path to the search location:

   ```
   <PATH:> \WINCE600\PUBLIC\COMMON\OAK\DRIVERS\ETHDBG\RNDISMINI\HOST\
   ```

10. Click **Next**.

    Now the `usb8023.inf` file will be used to install the RNDIS driver for the board.

11. You are prompted for the `usb8023.sys` file to install. This file is usually located at `\WINDOWS\system32\drivers` folder for Windows XP.

    If you experience installation problems, refer to:

    ```
    \WINCE600\PUBLIC\COMMON\OAK\DRIVERS\ETHDBG\RNDISMINI\HOST\ howto.txt
    ```

12. When the driver is correctly installed, the Microsoft Windows CE RNDIS virtual adapter from Network Connections will be displayed on the PC.

13. 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 with board

    Subnet mask: **255.255.255.0** //must match the boot configuration on the board

14. Follow the procedure in Section 6.2 to add the device to Platform builder; however, at Step 10, select “U” (for USB transport) rather than “D” (Ethernet transport).

Chapter 7
Building and Using the BSP Demo Applications

The BSP supports the following Windows Embedded CE 6.0 applications:

- Camera
- TV-Out
- Rotate

7.1 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.

7.1.1 Building the Camera Application

To build a Camera application, use these steps:

1. Create an **APP** folder under the following folder:
   `\Platform\iMX313DS\Src`
2. Create an empty **dirs** file under the following folder:
   `\Platform\iMX313DS\Src\App`
3. Copy the **CAMAPP** folder from the `\Support\App` folder to the `\Platform\iMX313DS\Src\App` folder.
4. In the Build menu:
   — **Copy Files to Release Directory After Build** selected
   — **Make Run-Time Image After Build** unselected
5. Select the Solution Explorer tab of the Platform Builder Solution Explorer window.
6. Expand **WINCE600** > **PLATFORM** > **iMX313DS** > **src** > **App**.
7. Right-click on the CAMAPP folder and select **Rebuild**.

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

7.1.2 Using the Camera Application

Table 7-1 describes the features and commands for the camera demonstration application.

Table 7-1 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** 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 horizontal.
Vertical Flipping	**View** > **Vertical Flipping** For flipping picture vertical.
Options	**View** > **Options** Change Resolution, Zoom, and Rotation in one command.
Capture a still picture	**Capture** (button at Capture the bottom of the application) The captured picture is the same resolution that that you 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.

7.2 TV-Out Application

The TV-Out application is `\windows\TVOUT.exe`. The TV-Out application toggles the Windows Embedded CE 6.0 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 Embedded CE 6.0 desktop to the TV-Out connector.
- **tvout.exe 1** sets the TV output format to NTSC and sends the Windows Embedded CE 6.0 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.

7.3 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 Embedded CE 6.0 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.

7.4 FM Radio Application

The FM Radio application provides an FM radio station receiver. The Si4702 on the 3-Stack platform uses the FM radio driver included in the BSP, and sends the sound to the audio output.

7.4.1 Building the FM Radio Application

To build an FM radio application, use these steps:

1. Create an **APP** folder under the following folder:
   `\Platform\iMX313DS\Src`
2. Create an empty **dirs** file under the following folder:
   `\Platform\iMX313DS\Src\App`
3. Copy the **FMAPP** folder from the `\Support\App` folder to the `\Platform\ iMX313DS\ Src\App` folder.
4. In the Build menu:
   — **Copy Files to Release Directory After Build** selected
   — **Make Run-Time Image After Build** unselected
5. Select the Solution Explorer tab of the Platform Builder Solution Explorer window.
6. Expand **WINCE600** > **PLATFORM** > **iMX313DS** > **src** > **App**.
7. Right-click on the FMAPP folder and select **Rebuild**.

   The `FMRadioApp.exe` file will be created in the workspace release directory.

7.4.2 Using the FM Radio Application

To use the application, click the FM Radio Application icon. Figure 7-1 illustrates the FM Radio's main window. The application window will fit the new client area dimensions after screen rotation, for example, from Portrait Screen to Landscape Screen. Table 7-2 describes the controls. Figures 6-2 through 6-5 illustrate the secondary dialogs.

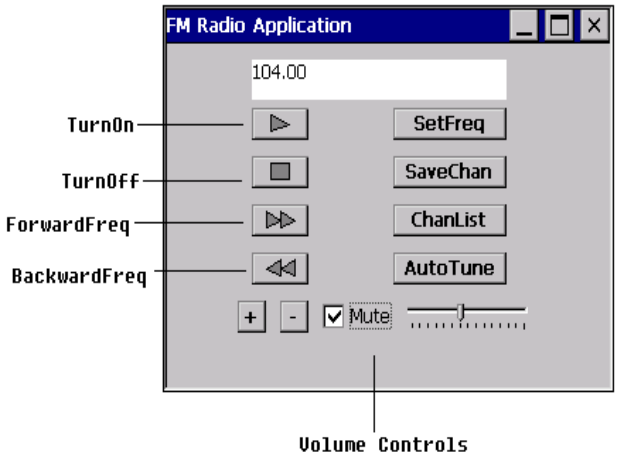

Figure 7-1 FM Radio Application Main Window

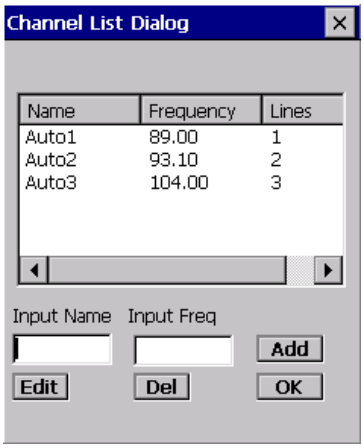

Figure 7-4 Channel List Dialog (Portrait Screen)

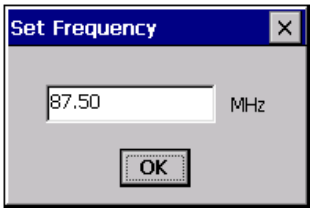

Figure 7-2 Set Frequency Dialog

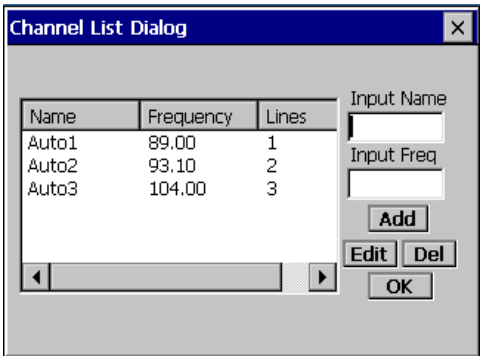

Figure 7-5 Channel List Dialog (Landscape Screen)

Figure 7-3 Save Channel Name Dialog

Table 7-2 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**. The input frequency range is from 87.5 megahertz to 108.00 megahertz.
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.

7.5 Accelerometer Demo Application

The Accelerometer Demo Application illustrates the automatic rotation of the screen when the device is rotated. To accomplish the rotation, the application uses the accelerometer driver included in the BSP, and reads the user-entered X, Y, and Z values.

7.5.1 Building the Accelerometer Demo Application

To build the Accelerometer Demo application, use these steps:

1. Download Windows Mobile 6 Professional and Standard Software Development Kits.
2. Install the SDK.
3. Open Microsoft Visual Studio 2005.
4. Click **File** > **Open** > **Project/Solution**, and select the project at:

   `SUPPORT\MX31\APPS\ACCAPP`

5. Right-click on the test and select **Rebuild**.

   The `Acctest.exe` file will be created in the workspace release directory.

7.5.2 Using the Accelerometer Demo Application

The following graphic illustrates the main window.

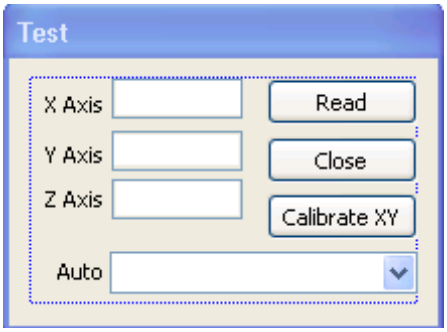

Figure 7-1 Accelerometer Demo Application Main Window

To use the application, follow these steps:

1. In the main window, note the X, Y, and Z Axis values.
2. To calibrate the X and Y Axis values, click **Calibrate XY**.
3. To select the X, Y, and Z Axis value update interval, click the down arrow next to Auto, and select the interval.
4. To cause the screen to rotate automatically when the device is rotated, click the down arrow next to **Auto**, and select **Auto Rotate**.

7.6 Wi-Fi Connection

This BSP includes Wireless LAN connectivity support through the CSR APM6628 chip included in the i.MX31 PDK.

To initialize the Wi-Fi connection, use these steps:

1. Go to the `\Windows` folder and double-click the **LoadDriveriMX31**.exe icon (Figure 7-3).

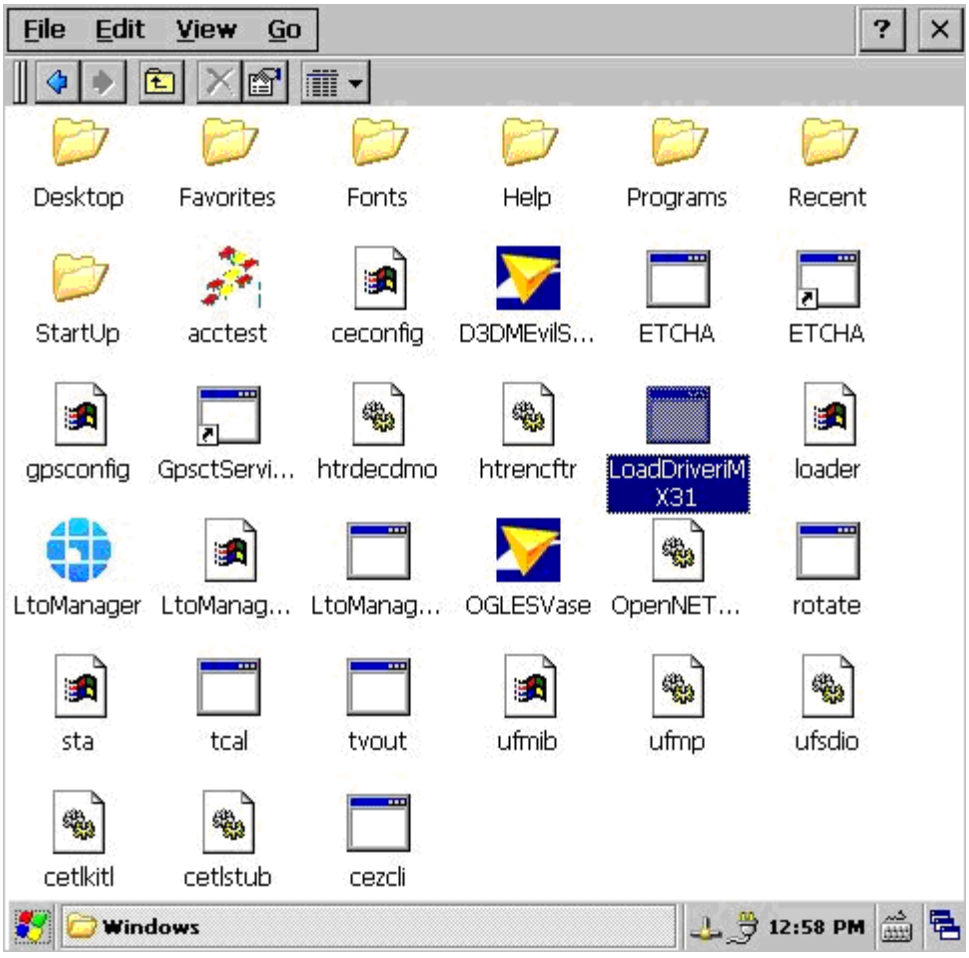

Figure 7-3 Wi-Fi Configuration Application

NOTE

The `LoadDriveriMX31` application should be executed only once. Additional attempts may cause WLAN driver instability.

The UFMPI dialog is displayed (Figure 7-4). This built-in application is included in the Windows Embedded CE 6.0 image by default.

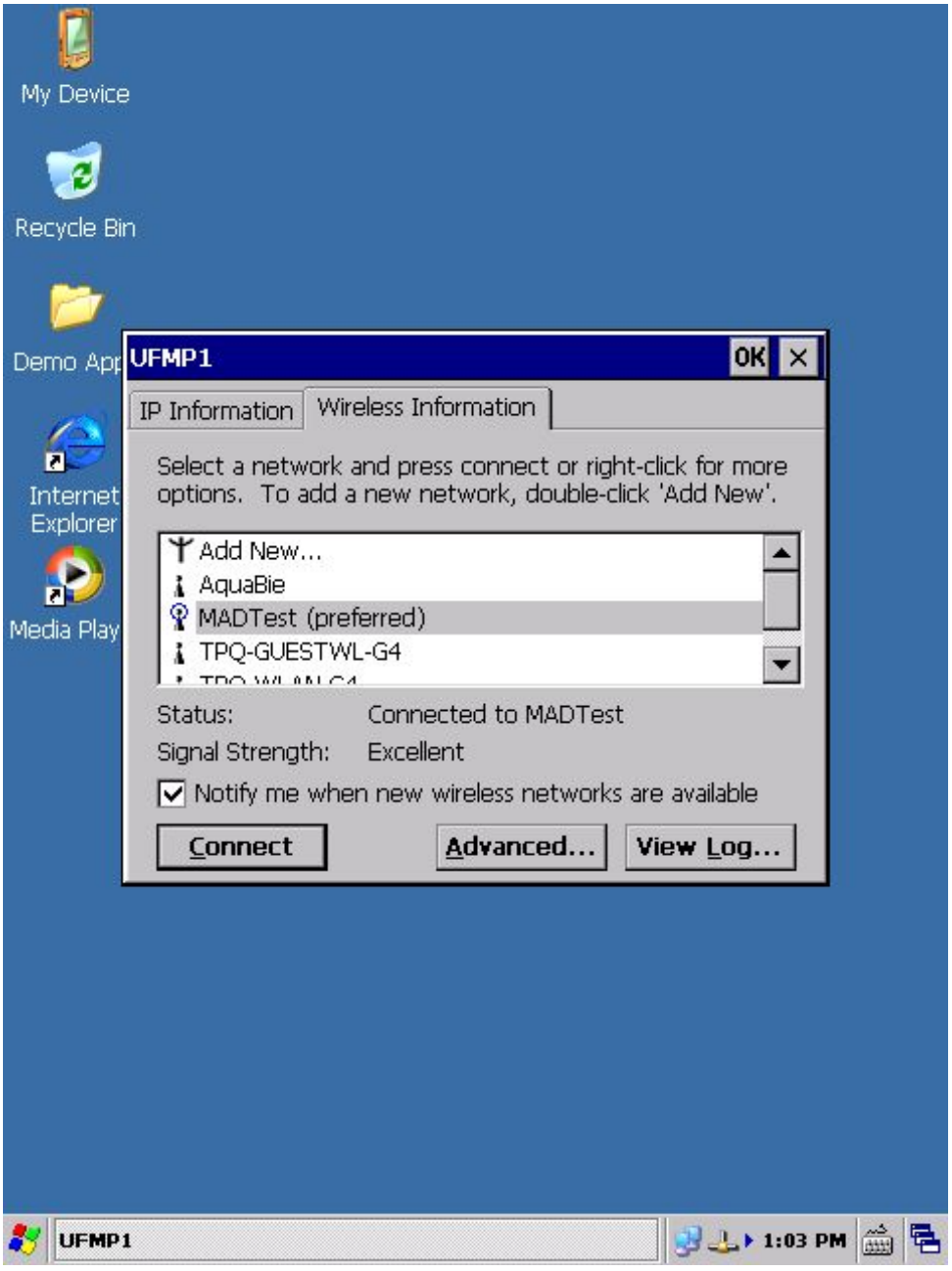

Figure 7-2 UFMPI Dialog

2. On the Wireless Information tab, select the Wireless Network that you want to connect to, and then click **Connect**.

   The Wireless Network Properties dialog is displayed for configuring the wireless connection settings (Figure 7-4).

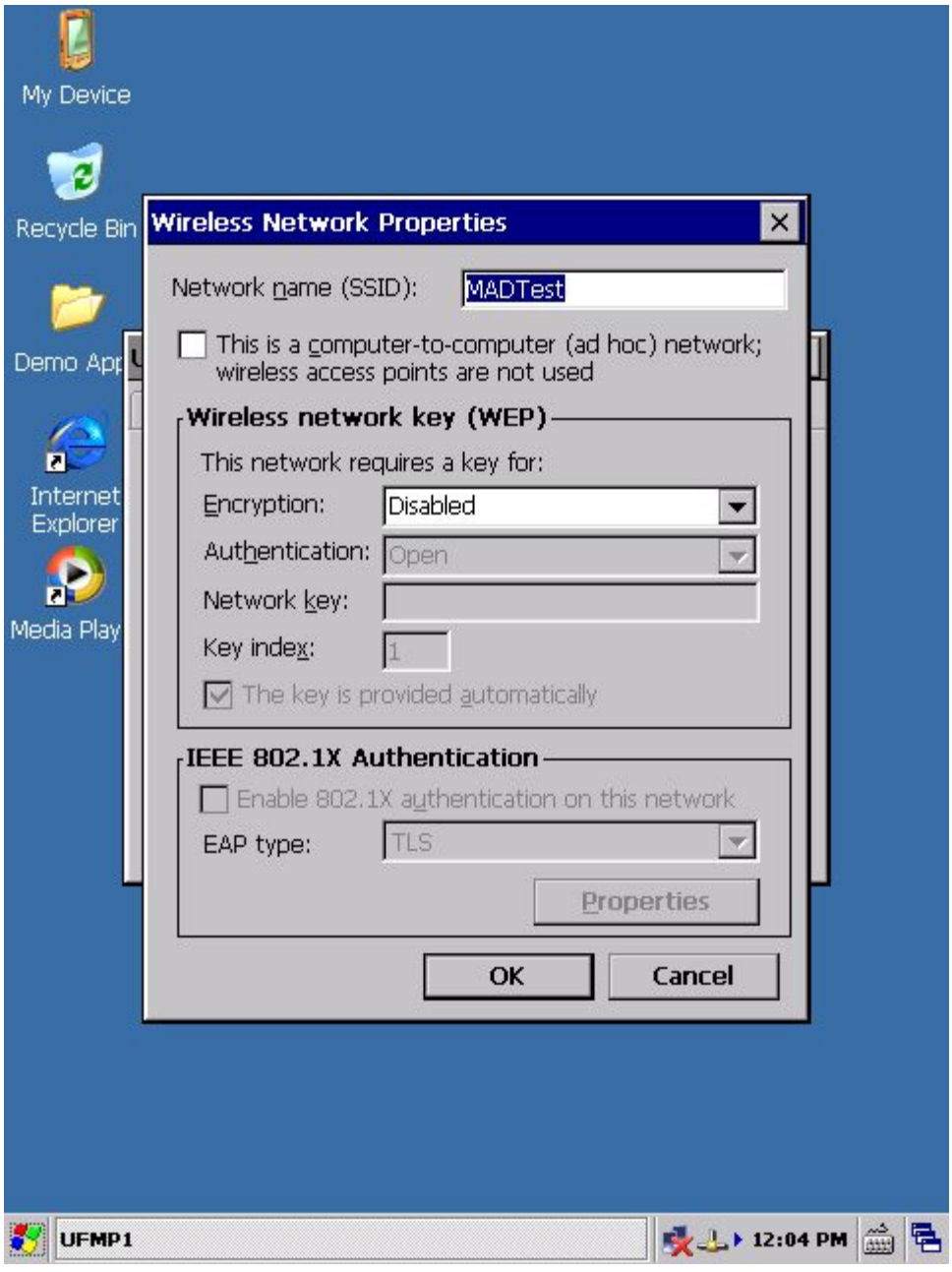

Figure 7-3 Wireless Network Properties

3. Select the needed settings, and then click **OK**.