i.MX28 EVK
Windows Embedded CE 6.0

User's Guide

Document Number: 924-76411
Rev. 2010.08
09/2010

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, Inc.
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
+1-303-675-2140
Fax: +1-303-675-2150
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 which 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.

ARM POWERED®

freescale™
semiconductor

Contents

About This Book

Chapter 1
BSP Installation

Chapter 2
BSP Contents and Organization

Chapter 3
Configuring OS Images

Chapter 4
Building OS Images

Chapter 5
Preparing for Downloading and Debugging

Chapter 6
Downloading and Debugging Images

Chapter 7
BSP Features

About This Book

This document is a user's guide for the Freescale i.MX28 EVK Windows Embedded CE 6.0 board support package (BSP). This document describes how to install the BSP and how to use Microsoft Platform Builder for Windows CE 6.0 to build, download, and debug OS images. Information on the configuration and usage of features included with the Freescale BSP is also provided.

Audience

This guide is intended for users of Microsoft Platform Builder who want to build and execute Windows CE 6.0 OS images based on the Freescale BSP. The audience also includes Windows CE application developers who want to leverage API interfaces provided by the Freescale BSP.

Organization

This document is organized into the following chapters:

Chapter 1	Describes how to install/uninstall the Freescale BSP.
Chapter 2	Describes the contents and organization of the Freescale BSP.
Chapter 3	Describes how to configure the Freescale BSP for building Windows Embedded CE 6.0 OS images.
Chapter 4	Provides instructions on how to build Windows Embedded CE 6.0 OS images using the Freescale BSP.
Chapter 5	Describes the preparation necessary for downloading and debugging OS images.
Chapter 6	Describes the procedures for downloading and debugging OS images.
Chapter 7	Provides details on the Freescale BSP features included in this release.

Suggested Reading

Additional information regarding Microsoft Windows CE and the i.MX28 can be found in the following documents:

- i.MX28 EVK Platform System HW User Guide
- Windows Embedded CE 6.0 BSP for i.MX28 EVK Reference Manual
- http://msdn.microsoft.com/embedded/windowsce
- Windows Embedded Training Resources: http://www.windowsembedded.com/training
- MCP Certification for Windows Embedded CE: http://www.windowsembedded.com/certification

Conventions

This document uses the following notational conventions:

- `Courier monospaced type` indicate commands, command parameters, code examples, expressions, datatypes, and directives.
- *Italic type* indicates replaceable command parameters.
- All source code examples are in C.

Definitions, Acronyms, and Abbreviations

The following list defines the acronyms and abbreviations used in this document.

Definitions and Acronyms

Term	Definition
API	Application programming interface
BSP	Board support package
CSP	Chip support package
DHCP	Dynamic host configuration protocol
EBOOT	Ethernet bootloader
EVB	Platform evaluation board
EVK	Platform evaluation kit
FAL	Flash abstraction layer
GDI	Graphics display interface
ICE	In-circuit emulator
IDE	Integrated development environment
IST	Interrupt service thread
IPU	Image processing unit
KITL	Kernel independent transport layer
LVDS	Low-voltage differential signaling
MAC	Media access control
OEM	Original equipment manufacturer
OAL	OEM adaptation layer
OS	Operating system
PQOAL	Production quality OEM adaptation layer
RVDEBUG	RealView debugger
RVI	RealView ICE
SDC	Synchronous display controller
SDRAM	Synchronous dynamic random access memory
SoC	System on a chip

References

The following documents were referenced to create this document.

- *Windows Embedded CE 6.0 Online Help*
- *i.MX28 Applications Processor Reference Manual*
- *i.MX28 EVK Windows Embedded CE 6.0 Release Notes*

- *i.MX28 EVK Windows Embedded CE 6.0 Reference Manual*

Chapter 1
BSP Installation

The board support package (BSP) is distributed as single Microsoft® Installer (.msi) file that includes support for the i.MX28 EVK platform. See the *i.MX28 EVK Windows Embedded CE 6.0 Release Notes* for additional instructions and information before installing and using BSP.

1.1 Installing Windows Embedded Tools and i.MX28 EVK BSP

To create the complete i.MX28 EVK development environment, install the Microsoft Windows Embedded CE 6.0 development tools and the Freescale i.MX28 EVK BSP as follows:

1. Install Visual Studio 2005 and the Windows Embedded CE 6.0 Platform Builder plugin from the installation discs. See the *i.MX28 EVK BSP Release Notes* on the Visual Studio and Platform Builder installation discs for installation instructions.

   NOTE

   When the platform builder installer asks for the operating system version, select the ARMV4I option. Install this option on the local hard drive. The other operating systems are not required.

   See the *i.MX28 EVK BSP Release Notes* for information about any additional Microsoft-supplied QFEs or Service Packs that also need to be installed.

2. If an earlier version of the Freescale i.MX28 EVK BSP has been previously installed, remove any existing BSP files.See Section 1.2, “Uninstalling the Freescale i.MX28 EVK BSP,” for instructions.
3. Download the Freescale i.MX28 EVK BSP and install the contents in the existing WINCE600 top-level folder. The MSI installer automatically creates the necessary subfolders so that all of the files are installed in the correct location.

1.2 Uninstalling the Freescale i.MX28 EVK BSP

This section describes how to uninstall BSP from the Windows Embedded CE 6.0 source code tree and Platform Builder development environment.

NOTE

Uninstalling the BSP removes all the files that were installed with the MSI installer. If any changes were made to these files, they are removed. Save the modified files before uninstalling the BSP.

The instructions to uninstall the BSP are as follows:

1. Close the Platform Builder.
2. Either rerun the original MSI installer and select the **Remove** option; or open the **Add or Remove Programs** Control Panel applet, select the Freescale BSP item, and then select **Remove**.
3. Manually remove the remaining BSP files and directories (some of these files and directories remain after the step2 because they contain object files or other generated files that are not part of the original BSP installation):

```
WINCE600\OSDesigns\iMX28-EVK-PDK1_9-SmallFootprint
WINCE600\OSDesigns\iMX28-EVK-PDK1_9-Mobility
WINCE600\OSDesigns\iMX28-EVK-PDK1_9-UUT
WINCE600\PLATFORM\COMMON\SRC\SOC\COMMON_FSL_V2_PDK1_9
WINCE600\PLATFORM\COMMON\SRC\SOC\MX28_FSL_V2_PDK1_9
WINCE600\PLATFORM\iMX28-EVK-PDK1_9
WINCE600\SUPPORT_PDK1_9
```

4. Manually remove the remaining BSP library files that are created after the BSP installation. To find the libraries, use Windows Explorer with the search name `*FSL_V2*` for the following library path and remove all the `LIB`, `PDB`, and `DEF` files found by the search:

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

1.3 Load Sample OS Design Solution

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

1. In the Platform Builder, select **File** > **Open** > **Project/Solution.**
2. Load the following files to select the i.MX28 EVK BSP sample workspace

```
WINCE600\OSDesigns\iMX28-EVK-PDK1_9-Mobility\iMX28-EVK-PDK1_9-Mobility.sln
WINCE600\OSDesigns\iMX28-EVK-PDK1_9-SmallFootprint\iMX28-EVK-PDK1_9-SmallFootprint.sln
WINCE600\OSDesigns\iMX28-EVK-PDK1_9-Mobility\iMX28-EVK-PDK1_9-UUT.sln
```

The process of loading this solution automatically loads the associated i.MX28 EVK BSP Catalog. Select the **Catalog Items View** tab to see all of the available OS solution catalog items.

Chapter 2 BSP Contents and Organization

The Freescale 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 i.MX28 EVK-based platforms. 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 BSP includes a set of directories and files that are installed into an existing Windows Embedded CE 6.0 source tree. The BSP directory structure follows the production-quality OAL (PQOAL) and production-quality driver (PQD) structure recommended by Microsoft.

The i.MX28 EVK system-on-a-chip (SoC) leverages a common Freescale ARM®-based platform architecture. This platform is found in a series of ARM-based SoCs available from Freescale. In order to leverage source code that is portable across multiple Freescale ARM-based SoCs, a common directory called `COMMON_FSL_V2_PDK1_9` is used to store shared OAL and driver components. This `COMMON_FSL_V2_PDK1_9` common directory appears in the chip support package and PQOAL directories described in following sections.

2.1 System-on-a-Chip Support Package (SoC)

The Freescale SoC directory contains chipset-level code that can be leveraged to develop platforms based on the i.MX28 EVK SoC and the PQOAL components customized for the i.MX28. The code and definitions in the SoC directory can be reused in a new platform design without modification. To keep the SoC sources platform agnostic, the source code in the SoC directory utilizes hardware abstraction routine 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.

2.1.1 Production Quality Driver (PQD)

Windows Embedded CE 6.0 supports PQD components that simplify and shorten the process of developing a driver. For more information on PQQL development concepts, see the section **Production-Quality Drivers** in the *Windows Embedded CE 6.0 Help*.

The following directories contain the SoC driver source code for the i.MX28 EVK:

```
WINCE600\PLATFORM\COMMON\SRC\SOC\COMMON_FSL_V2_PDK1_9
WINCE600\PLATFORM\COMMON\SRC\SOC\MX28_FSL_V2_PDK1_9
```

The SoC code in the `COMMON_FSL_V2_PDK1_9` directory is reusable across all Freescale ARM-based SoCs. The SoC driver code in the `MX28_FSL_V2_PDK1_9` directory is reusable across all platforms based on the i.MX28 EVK.

2.1.2 Production Quality OAL (PQOAL)

Windows Embedded CE 6.0 supports PQOAL components that simplify and shorten the process of developing an OAL. For more information on PQOAL development concepts, see the section **Production-Quality OAL** in the *Windows Embedded CE 6.0 Help*.

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.MX28 EVK are available in the following directories:

```
WINCE600\PLATFORM\COMMON\SRC\SOC\COMMON_FSL_V2_PDK1_9\OAL
WINCE600\PLATFORM\COMMON\SRC\SOC\MX28_FSL_V2_PDK1_9\OAL
```

The PQOAL code in the `COMMON_FSL_V2_PDK1_9\OAL` directory is reusable across all Freescale ARM-based SoCs. The PQOAL code in the `MX28_FSL_V2_PDK1_9\OAL` directory is reusable across all platforms based on the i.MX28 EVK.

2.2 i.MX28 EVK Platform Files

The i.MX28 EVK BSP provides direct support for the interfaces and peripherals in the i.MX28 EVK board. All the drivers and OAL content specific to the underlying hardware platform is located in the following directory:

```
WINCE600\PLATFORM\iMX28-EVK-PDK1_9
```

The i.MX28 EVK platform directory implements the hardware abstraction routines invoked by driver code in the Freescale SoC directory. Additionally, this directory implements certain aspects of the PQOAL that may need to be modified by the OEM for their specific platform.

2.3 Sample OS Design Solutions

Design solutions are used by Platform Builder to encapsulate the OS components and build options necessary for the Windows Embedded CE 6.0 tools to generate an OS image. Default solutions are included in the BSP in the following directory:

```
WINCE600\OSDesigns\iMX28-EVK-PDK1_9-Mobility
```

The following solution is included in the BSP that provides the workspace for building the MfgTool Windows Embedded CE 6.0 firmware run-time image:

```
WINCE600\OSDesigns\iMX28-EVK-PDK1_9-UUT
```

Another solution included in the BSP provides the starting point for the smallest functional Windows Embedded CE 6.0 run-time image:

```
WINCE600\OSDesigns\iMX28-EVK-PDK1_9-SmallFootprint
```

The solution was created using the Platform Builder **New** > **Project** feature utilizing the Custom Device Design Template. See the section **Creating an OS Design with the Windows Embedded CE OS Design Wizard** in the *Windows Embedded CE 6.0 Help* for more information about creating an OS solution.

2.4 Support Files

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

```
WINCE600\SUPPORT_PDK1_9
```

This directory is used to store applications and tests targeted for the supported platforms. Support files necessary to configure the development tools are also located in this directory.

Chapter 3 Configuring OS Images

Use the Platform Builder to select one of the two default build configurations provided in the BSP sample solution. These configurations control the type of OS image (debug or release) that is generated by the Platform Builder tools. In addition, the build configuration encapsulates all the platform environment variables and custom build instructions that are used during OS image creation. This chapter describes the configuration options available for the i.MX28 EVK BSP.

NOTE

> The sample solution provided in the i.MX28 EVK BSP is properly configured to generate a default image targeted for the i.MX28 EVK hardware. Do not modify any of the image configuration settings prior to building the BSP.

3.1 Image Build Type

The type of OS image generated by Platform Builder can be controlled using the **Build** > **Configuration Manager** menu item and choosing the appropriate **Active Solution Configuration** item from the drop-down list. The sample workspace provided with the i.MX28 EVK BSP provides the following two image build types:

- **Freescale_i_MX28_EVK_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_MX28_EVK_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.

NOTE

> To enable or disable the Standard toolbar select the **Tools** > **Customize** menu item. Choose the **Toolbars** tab and then select or deselect the **Standard** toolbar item.

For more information about the build types available for Windows CE, see the section **Build Configurations** in the *Windows Embedded CE 6.0 Help*.

3.2 BSP Environment Variables

The three types of BSP environment variables used to configure the BSP:

- BSP_NOXXX
- BSP_XXX
- IMGXXX

3.2.1 BSP_NOXXX Variables

The i.MX28 EVK BSP supports the Windows Embedded CE 6.0 dependency feature, which simplifies the BSP configuration process. The selection of certain SYSGEN components in the workspace (for example, SYSGEN_AUDIO) triggers the automatic selection of certain BSP drivers (for example, Audio). The drivers that have a SYSGEN dependency have a corresponding BSP_NOXXX variable defined in the platform batch file listed in the Table 3-1. These variables provide a means for excluding a driver from the OS image that might otherwise be included, due to a SYSGEN dependency.

```
WINCE600\PLATFORM\iMX28-EVK-PDK1_9\iMX28-EVK-PDK1_9.bat
```

NOTE

> It is not possible to remove the selection of the drivers by clicking the items in Catalog UI. Instead, set the corresponding BSP_NOXXX variables to 1 in **Environment** dialog, launched from Platform Builder menu **Project** > **Properties** > **Configuration Properties** > **Environment**.

Table 3-1 provides a summary of the BSP_NOXXX environment variables defined in the batch file.

Table 3-1. BSP_NOXXX Variables in Batch File

Variable Name	Description	Setting
BSP_NOAUDIO	Excludes support for SSI Audio driver	BSP_NOAUDIO=1 Excludes Audio driver support from the OS image despite the SYSGEN_AUDIO dependency.
BSP_NOBATTERY	Excludes support for SSI Battery driver	BSP_NOBATTERY=1 Excludes Battery driver support from the OS image despite the SYSGEN_BATTERY dependency.
BSP_NOCSPDDK	Excludes CSP driver development kit (CSPDDK) support. The CSPDDK is required for most BSP device drivers.	BSP_NOCSPDDK=1 Excludes CSPDDK from the OS image despite the SYSGEN_CEDDK dependency. Use this configuration only when building an OS design that does not include BSP device drivers.
BSP_NODISPLAY	Excludes support for Display driver	BSP_NODISPLAY=1 Excludes Display driver support from the OS image despite the SYSGEN_DISPLAY dependency.
BSP_NOKEYPAD	Excludes support for Keypad driver	BSP_NOKEYPAD=1 Excludes Keypad driver support from the OS image despite the SYSGEN_MININPUT dependency.
BSP_NONAND_FMD	Excludes support for NAND Flash driver	BSP_NONAND_FMD=1 Excludes NAND Flash driver support from the OS image despite the SYSGEN_FLASHMDD dependency.
BSP_NONLED	Excludes support for NLED driver	BSP_NONLED=1 Excludes NLED driver support from the OS image despite the SYSGEN_NLED dependency.

Table 3-1. BSP_NOXXX Variables in Batch File (continued)

Variable Name	Description	Setting
BSP_NOTOUCH	Excludes support for Touch driver	BSP_NOTOUCH=1 Excludes Touch driver support from the OS image despite the SYSGEN_TOUCH dependency.
BSP_NOUSB	Excludes support for all USB drivers	BSP_NOUSB=1 Excludes all USB drivers support from the OS image despite the SYSGEN_USB and SYSGEN_USBFN dependency.

NOTE

The opposite setting of BSP_NOXXX = 1 is either BSP_NOXXX =, which is set in the batch file by default, or no definition of the BSP_NOXXX variable. Thus, the driver is included by SYSGEN selection.

Although the drivers listed above have direct SYSGEN dependency, some of them may not be automatically selected by SYSGEN dependency because of the following possibilities:

1. Driver may have multiple subordinate selections, for example there are two selections SSP0 and SSP1 for SD Controller.
2. Driver may depend on another driver, for example Touch driver depends on the LRADC driver.
3. Driver may conflict with other drivers, for example NAND driver conflicts with SSP0 SD controller driver.

3.2.2 BSP_XXX Variables

The i.MX28 EVK BSP uses the BSP_XXX variables defined in the catalog to configure the drivers that can not be automatically selected by SYSGEN dependency. In this case, use the Platform Builder Catalog UI to select or deselect drivers by clicking the catalog items. The *i.MX28 EVK Windows Embedded CE 6.0 Reference Manual* describes the BSP_XXX variables for individual drivers.

NOTE

Selecting the desired drivers in the catalog graphical user interface can not be always successful. Users can see a small red **x** for the cases where dependency or conflict involved in the selection. In this case, right-click the catalog item and see the details in **Reasons for Exclusion of Item**. Use the information shown to select the required SYSGEN components or drivers and to deselect conflicting components.

When creating a custom workspace using the Platform Builder wizard, manually select the desired drivers that are not automatically selected by SYSGEN dependency using the BSP catalog items under **Third party** > **BSP** > **Freescale i.MX28 EVK: ARMV4I**.

3.2.3 IMG_XXX Variables

The IMG_XXX variables are used to control the Make Run-Time Image procedure. View and configure the variables in the **Environment** dialog as follows:

1. Open the sample solution for the i.MX28 EVK BSP.
2. From the **Project** menu, choose **Properties**.
3. Expand **Configuration Properties** (if necessary) and select the **Environment** item.

Table 3-2 provides a summary of the IMG_XXX environment variables available on i.MX28 EVK BSP.

Table 3-2. IMG_XXX Variables

Variable Name	Description	Setting
IMGSDMMC	EBOOT and OS build files uses to determine if images are targeted for SD/MMC card	IMGSDMMC = 1 Link EBOOT and OS images for SD/MMC card

3.3 Catalog Environment Variables

The i.MX28 EVK BSP utilizes the Platform Builder Windows CE Catalog to configure BSP components in the OS design. The *i.MX28 EVK Windows Embedded CE 6.0 Reference Manual* describes the environment variables associated with each BSP features exposed in the i.MX28 EVK BSP Catalog.

Chapter 4
Building OS Images

This chapter provides instructions for building Windows Embedded CE 6.0 OS images using the BSP.

4.1 Building the Freescale SoC Libraries

The Freescale SoC libraries that support the i.MX28 EVK are generated during the **Build** > **Advanced Build Commands** > **Sysgen** or **Build** > **Advanced Build Commands** > **Build Current BSP and Subprojects** build procedure. Windows CE ships with pre-built SoC libraries for various ARM processors, but the libraries for the i.MX28 EVK must be built from the sources since they are not included with the standard Microsoft distribution.

The provided sample OS design solution is preconfigured to build the required Freescale SoC. That is, the sample i.MX28 EVK OS design solution automatically builds the i.MX28 SoC sources.

The steps to build the Freescale SoC libraries are as follows:

1. Open the sample solution.
2. Select the desired build type. See Section 3.1, “Image Build Type.”
3. Select a **Sysgen** build if a Sysgen operation has never been performed. If a Sysgen operation has already been performed, it is only necessary to **Build Current BSP and Subprojects**, which rebuilds the Freescale SoC libraries.

For a release build type, the SoC libraries are located in following directory:

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

For a debug build type, the SoC libraries are located in following directory:

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

4.2 Building Run-Time Images

See the standard procedures described in the Platform Builder documentation for the remaining steps to build an OS image. For more information, see the section **Building a Run-Time Image** in the *Windows Embedded CE 6.0 Help*.

4.2.1 Building the BSP for the First Time

The steps to build the BSP for the first time are as follows:

1. Open the sample solution.
2. Select the desired build type. See Section 3.1, “Image Build Type.”
3. Using the **Build** > **Global Build Settings** menu, configure the build options as follows:
   - Select **Copy Files to Release Directory After Build**
   - Select **Make Run-Time Image After Build**
4. Select **Build** > **Advanced Build Commands** > **Sysgen** to start the build process.

For a release build type, the OS image files are located in the following directory depending on the OS design being used:

```
WINCE600\OSDesigns\iMX28-EVK-PDK1_9-Mobility\RelDir\Freescale_i_MX28_EVK_ARMV4I_Release
```

For a debug build type, the OS image files are located in the following directory:

```
WINCE600\OSDesigns\iMX28-EVK-PDK1_9-Mobility\RelDir\Freescale_i_MX28_EVK_ARMV4I_Debug
```

4.2.2 Clean Build for the BSP

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

A clean build of all the Freescale BSP components are performed as follows:

1. Select **Copy Files to Release Directory After Build**.
2. Deselect **Make Run-Time Image After Build**.
3. Select **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.

4.2.3 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. Perform a **Sysgen** again only when components are added or removed from the **OS design**. Instead, perform an incremental build of the Freescale BSP components to build an updated OS image as follows:

1. Open a solution.
2. Select the desired build type. See Section 3.1, “Image Build Type.”
3. Using the **Build** > **Global Build Settings** menu, configure the build options as follows:
   — Select **Copy Files to Release Directory After Build**
   — Select **Make Run-Time Image After Build**
4. Select **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.

Chapter 5 Preparing for Downloading and Debugging

The target and development workstation must be properly configured and initialized before OS images can be downloaded and executed. This chapter discusses the steps required to prepare the target and development workstation so that the Platform Builder can be used to download and debug images on the target.

5.1 Serial Debug Messages

Serial debug messages are used by the boot loader and OS images to report status and error information. Additionally, the boot loader uses serial input to allow user interaction. This section describes the configuration of the desktop workstation and Freescale BSP to support serial debug messages.

5.1.1 Desktop Workstation Serial Debug Port

Any terminal emulation application can be used to display messages send from the serial port of the target. Configure the terminal application with the following communications parameters:

- Bits per second—115200
- Data bits—8
- Parity—None
- Stop bits—1
- Flow control—None

5.1.2 Target Serial Debug Port

The i.MX28 EVK BSP is configured to use internal DUART routed to the connector J83 on i.MX28 EVK board for serial debug messages. Connect a standard serial cable between J83 DEBUG UART on the i.MX28 EVK board and the development workstation.

5.2 Board Configuration

5.2.1 i.MX28 EVK Board Configuration

Table 5-1 shows the switch and jumper configuration required for proper operation of the BSP on the i.MX28 EVK board.

Table 5-1. Switch and Jumper Configuration

Jumper/Switch	Recommended Setting	Comments
S1	REG	Used to select battery source: - REG: simulated battery by regulator - BATT: real battery (either J85 or J86 needs to be connected with a real battery)
S3	ON	Used to switch the VBUS power of USB connector J82: - ON: switch power on - OFF: switch power off
S16	WALL 5V	Used to select i.MX28 VDD5V source: - WALL 5V: i.MX28 VDD5V sources from WALL 5V - USB 5V: i.MX28 VDD5V sources from USB VBUS 5V
S17	JTAG OFF	Used to enable JTAG (no implication on WinCE BSP)
J94	5V (short 1-2)	Used to select power source of REG_3V3 and REG_1V8, which are used for powering up some components on the board: - 5V: source from WALL 5V - BATT: source from battery either real or simulated

5.2.2 i.MX28 EVK Boot Mode Settings

The switch S2 on i.MX28 EVK board selects i.MX28 Boot Mode. Table 5-2 shows the configuration in detail.

Table 5-2. S2 Boot Mode Select

Boot Mode	S2 (1-2-3-4) Setting
SD/MMC Boot	1-0-0-1
USB Recovery	0-0-0-0

Value ‘1’ of the setting refers to the side that labels number 1, 2, 3 and 4 and value ‘0’ refers to the side that labels only dots.

5.3 EBOOT Installation and Configuration

The Ethernet boot loader (EBOOT) is used to download and execute OS images. EBOOT is typically programmed in the storage memory on the target hardware and executes immediately out of reset. Initially, the target hardware does not have EBOOT resident in the storage memory. Furthermore, the target hardware has non-volatile storage for the EBOOT network configuration (DHCP/static, MAC address, and so on) that must be initialized before using the boot loader with the Platform Builder. This section describes the procedure for building, installing, and configuring EBOOT.

5.3.1 Building EBOOT Image

See Section 4.2, “Building Run-Time Images,” to build a retail OS image. During the process of building the OS image, the EBOOT image is created. To rebuild EBOOT (if required):

1. Select the **Solution Explorer** tab of the Platform Builder window.
2. Expand **iMX28-EVK-Mobility-PDK1_9** > **F:/WINCE600** > **PLATFORM** > **iMX28-EVK-PDK1_9** > **src** > **BOOTLOADER**.

NOTE

> The drive letter **F** may vary according to the actual Platform Builder installation path.

3. Right-click on the **EBOOT** folder and select **Rebuild**.

5.3.2 Installing EBOOT to SD/MMC Card

The steps to load EBOOT to the SDRAM using USB recovery mode are as follows:

1. Configure the target and desktop workstation for serial debug messages. See Section 5.1, “Serial Debug Messages,” for more information.
2. Connect the Ethernet cable to i.MX28 EVK board Ethernet port (the lower one).
3. Connect the USB cable between the i.MX28 EVK board USB OTG connector J82 and development workstation.
4. Plug the SD/MMC card into slot U42, which is on the bottom side of i.MX28 EVK board.
5. Prepare the target hardware by following the instructions in Section 5.2.1, “i.MX28 EVK Board Configuration”. See Section 5.2.2, “i.MX28 EVK Boot Mode Settings,” to power on the board in USB recovery mode.

NOTE

> Normally, a press on S4 (RESET) should power on or reset the system. But if not, another press on S5 (POWER) is required.

6. Open a command prompt, change directory to `WINCE600\SUPPORT_PDK1_9\TOOL\iMX28-EVK\SBIMAGE` and run the following command to download and launch EBOOT:

```
sb_loader /f eboot.sb
```

NOTE

> To boot from HAB_ENABLE silicons, the set of boot images with “_ivt” suffix should be used. The command line becomes “`sb_loader /f eboot_ivt.sb`” for HAB_ENABLE case.

7. Switch over to the terminal emulation application and the debug message **Press [ENTER] to download now or [SPACE] to cancel** appears.
8. Press the space bar and EBOOT configuration menu appears.

9. Select **ENET** in configuration menu and **Select Ether Device**.
10. Specify an Ethernet MAC address by selecting the **Set MAC Address** menu option and then entering the 12-digit hexadecimal MAC address.

NOTE

> Lables on the left side of the i.MX28 EVK board describes the MAC addresses for the Ethernet ports.

11. Enable **DHCP** option to avoid to specifying a static IP or static subnet mask. Use the **IP Address** and **Set Subnet Mask** menu options, if static networking parameters are required.
12. After the desired network configuration appears in the EBOOT menu, select the **Save Settings** menu option to save the configuration to SD/MMC card.
13. From the **File** menu of the Platform Builder choose **Open Workspace**.
14. Select eboot.msb in Platform Builder project Property Pages (Alt+F7).

NOTE

> Platform Builder does not recognize .msb file type, so eboot.msb is not shown in **Target file name for debugger** drop list of Property Pages. Manual entry of "eboot.msb" (or "eboot_ivt.msb" for HAB_ENABLE case) in the text box performs if the file exists in flat release directory.

15. Follow the steps in Section 5.4, "Configuring Ethernet Connection for Downloading and Debugging," to establish an Ethernet connection between the target and the Platform Builder.
16. From the **Target** menu, select **Attach Device**.
17. Switch over to the terminal emulation application, and specific SD/MMC according to the following prompt message:

```
Specify EBOOT destination [1 = NOR, 2 = NAND, 3 = SD/MMC]:
```

18. Press the **y** key to confirm the format of the card.

```
We must format storage before writing image to it, do you want to continue? [Y/N]
```

NOTE

> The required card capacity is larger than 64 Mbytes for storing boot images. The remaining space is reserved for FAT32 partition, which can be created later by formatting the card on Windows XP system.

19. EBOOT programs the image into SD/MMC card and provides status using serial debug messages. After the programming is complete, the following messages appears:

```
INFO: Update to EBOOT completed successfully.
Reboot the device manually...
SpinForever...
Do you want to reset [Y/N]
```

20. Use the instructions provided in Section 5.2.2, "i.MX28 EVK Boot Mode Settings," to configure the EVK board for boot from the SD/MMC card.
21. Reset the target hardware. If new EBOOT messages appear on the terminal, EBOOT has been properly programmed into SD/MMC card.

5.4 Configuring Ethernet Connection for Downloading and Debugging

The steps to configure an Ethernet connection for downloading and debugging images are as follows:

1. From the Platform Builder **Target** menu, choose **Connectivity Options**.
2. Choose **Kernel Service Map**.
3. In the **Target Device** box, choose a target device.

NOTE

> If a connection to a target device is already configured, the settings associated with the connection to the target device appear in the **Download** box and the **Transport** box.

4. In the **Download** box, choose **Ethernet** as the download service.
5. Launch EBOOT on the target. After EBOOT initialization completes, BOOTME messages appear on the serial debug output. Note the device name created by EBOOT on the serial debug output.

NOTE

> If EBOOT is already programmed into the target, a hardware reset starts EBOOT execution. If EBOOT is not programmed into the target, see Section 5.3.2, “Installing EBOOT to SD/MMC Card”, for steps 1 ~ 6 to launch EBOOT.

6. To the right of the **Download** box, choose **Settings**. The device name of the target should appear in the **Active Devices** box.
7. Select the target from the **Active Devices** box, and then choose **OK**.
8. In the **Transport** box, choose **Ethernet** as a kernel transport.
9. To the right of the **Transport** box, choose **Settings**.
10. Check **Use device name from bootloader** and then choose **OK**.
11. If the run-time image includes support for the kernel debugger stub, KdStub, from the **Debugger** box, choose **KdStub**. If the run-time image does not include support for a debugger, from the **Debugger** box, choose **None**.
12. Choose **Core Service Settings**.
13. To instruct the Platform Builder to download a run-time image each time the Platform Builder connects with the target device, under **Download Image**, choose **Always**.
14. Select **Enable KITL on device boot**.
15. Select **Clear memory on soft reset**.
16. Select **Enable access to desktop files**.
17. Choose **Apply**.
18. Choose **Close**.

Chapter 6
Downloading and Debugging Images

This chapter describes the procedures for downloading and debugging OS images on the i.MX28 EVK board. It is assumed that the steps to build an OS image have been followed and that development hardware has been configured prior to attempting a download and debug session.

6.1 Downloading OS Image to SDRAM Using EBOOT

The steps to download an image into the SDRAM of the target are as follows:

1. If EBOOT is not resident on the target, see Section 5.3, “EBOOT Installation and Configuration”, to program EBOOT into the target.
2. Prepare the target hardware by following the instructions in Section 5.2.1, “i.MX28 EVK Board Configuration”. Configure the EVK board for boot from SD/MMC card.
3. Open the desired workspace in the Platform Builder.
4. Deselect **Project** > **Properties** > **Configuration Properties** > **Build Options** > **Write Run-time Image to Flash Memory**.
5. In **Project** > **Properties** > **Configuration Properties** > **Environment**, remove the IMGSDMMC environment variable if it exists.
6. Build the image. See the Chapter 4, “Building OS Images.”
7. Reset the EVK board to launch EBOOT on the target.
8. If a target device connection has not been created in the Platform Builder, see the Section 5.4, “Configuring Ethernet Connection for Downloading and Debugging,” to establish a connection.
9. From the Platform Builder **Target** menu, select **Attach Device** to download.

6.2 Downloading OS Image to SD/MMC Card Using EBOOT

The steps to download an image into the SD/MMC card of the target are as follows:

1. If EBOOT is not resident on the target, see the Section 5.3, “EBOOT Installation and Configuration,” to program EBOOT into the target.
2. Prepare the target hardware by following the instructions in Section 5.2.1, “i.MX28 EVK Board Configuration.” Configure the EVK board for boot from SD/MMC card.
3. Open the desired workspace within the Platform Builder.
4. Within **Build** > **Properties** > **Configuration Properties** > **Environment**, set the IMGSDMMC environment variable if it does not exist.
5. Build the image. See the Chapter 4, “Building OS Images.”
6. Reset the EVK board to launch EBOOT on the target.
7. If a target device connection has not been created within the Platform Builder, see the Section 5.4, “Configuring Ethernet Connection for Downloading and Debugging,” to establish a connection.
8. From the Platform Builder **Target** menu, select **Attach Device** to download.
9. After the download is complete, switch over to the terminal emulation application.

10. EBOOT downloads the image temporarily into SDRAM and is ready to begin the Flash programming procedure.
11. EBOOT programs the image to a SD/MMC card and provides status using serial debug messages. After the programming is complete, the following messages appear:

```
INFO: Update of NK completed successfully.
Reboot the device manually...
SpinForever...
```

6.3 Running OS Image from SD/MMC Card Using EBOOT

The steps to execute an OS image from a SD/MMC card that has been previously programmed using the procedure described in Section 6.2, “Downloading OS Image to SD/MMC Card Using EBOOT,” are as follows:

1. Reset the EVK board to launch EBOOT on the target.
2. Switch over to the terminal emulation application and the debug message, **Press [ENTER] to download now or [SPACE] to cancel** appears.
3. Press the space bar and the EBOOT configuration menu appears.
4. Continue selecting the **Select Boot Device** option of the EBOOT menu until SD/MMC is selected.
5. Specify the desired boot delay using the **Boot Delay** menu option.
6. Save the configuration using the EBOOT menu.
7. Reset the EVK board to launch EBOOT again. EBOOT automatically loads the OS image from SD/MMC to RAM and executes it after the specified boot delay.

NOTE

If the EVK board is configured for active KITL, the image waits for a Platform Builder connection before booting the OS image. Because the OS image is already resident in Flash memory, it is necessary to reconfigure the Platform Builder to jump directly to the image. Selecting **Target** > **Connectivity Options** > **Core Service Settings** > **Download Image: Never (jump to image)**.

Chapter 7 BSP Features

7.1 Supported Features and Device Drivers

Details of the drivers and the kernel support provided in the i.MX28 EVK BSP are described in the *Windows Embedded CE 6.0 BSP for i.MX28 EVK Reference Manual*. This document contains detailed information regarding the requirements, implementation, and testing for each i.MX28 EVK BSP features.