i.MX Advanced Toolkit Standard Version

User's Guide

Document Number: 926-77203
Rev. 1.66
02/2009

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.

Contents

About This Book

This guide explains how to use the Advanced Toolkit (ATK) to perform Flash operations on a chip application development board.

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.

This document contains the following chapters.

Chapter 1 Describes installation and configuration.

Chapter 2 Describes the ATK tools.

Chapter 3 Illustrates and describes the operational procedures.

Chapter 4 Provides instructions for using the Flash tool.

Chapter 5 Explains how to convert image formats from binary to S-record, S-record to binary, and ELF to binary.

Chapter 6 Provides information about the memory initialization file format, configuring the ATK for high-speed USB on the i.MX27 TO2, and the purpose and method for checking the BBT/BI swap in the flash tool.

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.

Definitions, Acronyms, and Abbreviations

The following list defines the abbreviations used in this document.

ATK Advanced ToolKit

Device Program A program that includes the RAM kernel and Flash library. The device program runs in external RAM and communicates with the host machine to execute specific operations

Chapter 1 Introduction

The Advanced ToolKit (ATK) is a graphical user interface application for use in an i.MX platform for development and validation.

The ATK provides the following features:

- A Flash tool for downloading, programming, dumping, and erasing images in Flash memory to the i.MX board
- An image conversion tool for converting the following image file formats: binary to S-record, S-record to binary, and ELF to binary

The ATK runs on Windows platforms. You will need a PC with Windows 2000 SP4 or Windows XP.

NOTE

The ATK is used for application development and is not a mass production tool.

Software installation includes the tools listed in Table 1-1. To use a USB connection, you must also install the USB driver.

Table 1-1 Installation Requirements for Tools in ATK

Installations Tools	Standard ATK Tool installation package	USB Driver (Optional for USB connection)
Flash Tool	Yes	Yes
Image Convert	Yes	

Table 1-2 identifies the supported flash types for each board.

Table 1-2 Supported Flash Types

HW	NOR	NAND	MMC/SD
i.MX31 ADS board	S71WS256ND0	K9K1G08U0B	Supported
i.MX27 ADS board	S71WS256ND0	K9K1G08U0B	Not Supported
i.MX27 3-Stack board	Not Supported	K9F2G08R0A	Not Supported
i.MX31 3-Stack board	Not Supported	K9F2G08R0A	Supported
i.MX32 ADS board	S71WS256ND0	K9K1G08U0B	Supported
i.MX35 TO1/TO2 3-Stack board	SG29GL215	K9LAG08U0M	Supported
i.MX37 3-Stack board	Not Supported	K9LBG08U0M	Supported
i.MX51 TO1 3-Stack board	Not Supported	K9GBG08U0M	Supported
i.MX51 TO2 3-Stack board	Not Supported	MT29F32G08QAA	Supported
i.mx25 3-Stack board	Not supported	K9LAG08U0M	Supported

1.1 Installing the Standard ATK Package

Installation requires extracting the files to a temporary folder, and then executing a setup file.

CAUTION

Before installing the ATK, you must first remove any previous installation, and then reboot your PC. To do so:

From the Start Menu, point to **Programs** > **Advanced Toolkit**, and then click **Uninstall Advanced Toolkit**.

OR

From the Start Menu, point to **Settings** > **Control Panel** > **Add or Remove Programs**, and then select to uninstall the ATK.

You must reboot the PC after removing a previous installation. If you do not reboot at that time, but instead continue with the installation and its subsequent reboot, the operation will fail. If the operation fails, uninstall again, and then reboot.

To install the standard ATK tool, use these steps:

1. Extract the ATK package and place it in a temporary folder.

2. From the top level folder, execute the following file:

   ```
   FSL_ATK_TOOL_STD_INSTALL_<version_number>.exe
   ```

   After executing this file, you do not need to reboot.

3. From the Start Menu, point to **Programs**, then to **Advanced ToolKit**, and then click **Advanced ToolKit**.

The Advanced ToolKit configuration screen is displayed. At this point, the USB Drivers will also be installed. Proceed to the next chapter to configure the hardware.

1.2 (Optional) Installing and Uninstalling the USB Drivers

During the installation procedure, the USB drivers should have installed automatically, following the software tool installation. If the drivers did not install, run the `wd_install_4NT.bat` file, which is located in this folder:

```
<TOOL_INSTALL_PATH>/windriver/
```

To uninstall the USB driver, either uninstall the ATK, or run the `wd_uninstall_4NT.bat` file, which is located in the following folder:

```
<TOOL_INSTALL_PATH>/windriver/
```

You can also use this program to uninstall a previous version of USB drivers.

To verify that the USB driver is ready to use, navigate to the Device Manager, Jungo. You should see subfolders similar to those in Figure 1-1.

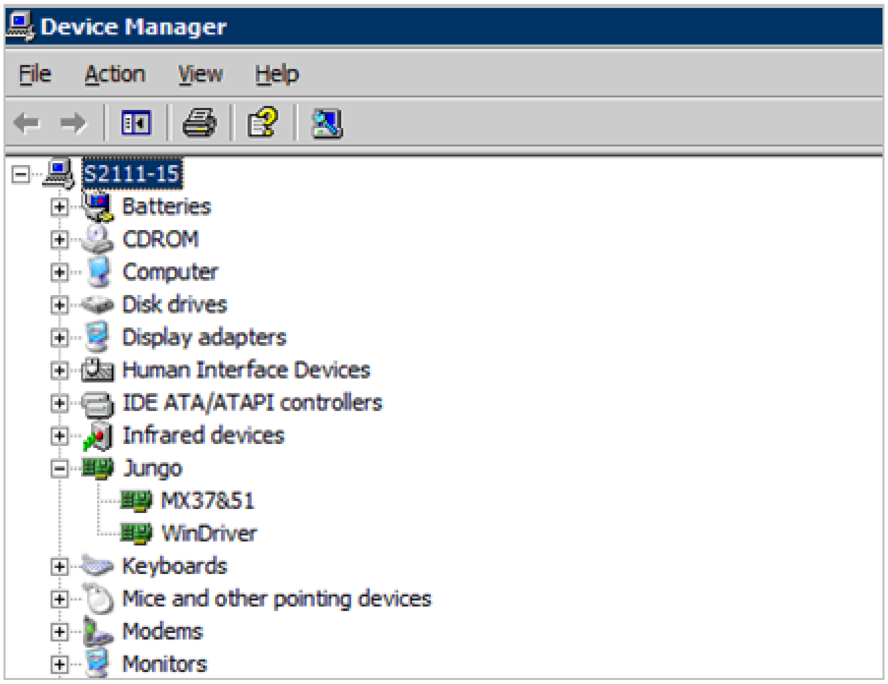


Figure 1-1 USB Interface

Chapter 2
Configuring the Hardware

Hardware configuration requires setting the board switches, and then connecting the board to the host computer.

2.1 Setting the Board Switches

Follow the instructions below for your board type.

NOTE

To program RedBoot on the pilot board, J28 must be open.

2.1.1 i.MX31/i.MX32 ADS Board

To set the jumper pins, use these steps:

4. Change the JP22 jumper (**CLK SEL**) on the CPU board to select **FPM clock** as the PLL reference clock.

NOTE

Do not select CKIH as the PLL reference clock.

5. By default, jumper pins 1 and 2 are connected together; change the jumper state to connect pins 2 and 3 together.
6. Enable the UARTC PORT on the base board, by setting switch 4 on the SW1 switch box to ON.
7. Connect the PC COMx port to the UARTC port (the top one).
8. Use the information in Table 2-1 to set the SW2 switches for the desired boot mode.

Table 2-1 Configure Boot Mode on SW2 Switch

SW2	1	2	3	4	5
Internal boot (bootstrap mode for programming)	ON	ON	ON	ON	ON
External boot from Flash	ON	ON	OFF	ON	OFF

NOTE

The MX31 T01 chip does not support connecting the COM port using a USB UART cable. This is because the UART is set to use Odd parity with two stop bits, which is not supported when using the USB UART cable in ROM codes.

2.1.2 i.MX31 3-Stack Board

Use Table 2-2 to select the settings on the Debug board for the desired boot mode.

Table 2-2 Switches on the i.MX31 3-Stack Board

Boot Mode Device	SW5	BOOT4 SW6	BOOT3 SW7	BOOT2 SW8	BOOT1 SW9	BOOT0 SW10
Internal boot (bootstrap mode for programming)	0	0	0	0	0	0
8-bit NAND Flash (2KB page) Ext	0	1	0	0	0	0

2.1.3 i.MX27 3-Stack Board

To configure the i.MX27 3-Stack board, use the information in Table 2-3 to set the SW5-SW10 dips on the Debug board for the desired boot mode. The rest of the chip switches should remain as they were set by default.

Table 2-3 Switches on the i.MX27 3-Stack Board

Boot Mode Device	BOOT3 SW7	BOOT2 SW8	BOOT1 SW9	BOOT0 SW10
Internal boot (bootstrap mode for programming)	ON	ON	ON	ON

2.1.4 i.MX35 TO1 3-Stack Board

2.1.4.1 Prerequisites

Ensure that the chip has been fused in the "non-security" or "engineering" mode.

2.1.4.2 Hardware Setting for Download Mode

To configure the i.MX35 3-Stack board, use the information in Table 2-4 to set the switches for the Debug and Personality boards.

Table 2-4 Switches on the i.MX35 TO1 3-Stack Board

Green Debug Board		Personality Board	
SW9	SW10	SW2	
		Dip1	Dip2
OFF	OFF	ON	ON

2.1.5 i.MX35 TO2 3-Stack Board

2.1.5.1 Prerequisites

Ensure that the chip has been fused in the "non-security" or "engineering" mode.

2.1.5.2 Hardware Setting for Download Mode

To configure the i.MX35 TO2 3-Stack board, use the information in Table 2-5 to set the switches for the Debug and Personality boards.

Table 2-5 Switches on the i.MX35 TO2 3-Stack Board

Green Debug Board		Personality Board	
SW9	SW10	SW2	
		Dip1	Dip2
ON	ON	OFF	OFF

2.1.6 i.MX37 3-Stack Board

2.1.6.1 Prerequisites

Check that these prerequisites have been met:

1. Ensure that the chip is fused in "non-security" or "engineering" mode.
2. Ensure that the UART fix that is identified in MCIMX37CE.pdf (Errata Number 20) has been programmed to I2C EEPROM by the ICE program. To download the fix, use these steps:
   - Modify the file path of `uart_dcd.bin in Marley_init_IIC.inc`: as follows:

     ```
     readfile,raw,gui
     "D:\Project\Marley\Testcode\BIN\uart_dcd.bin"=0x41000000
     ```
   - In RVDS, load `Marley_init_IIC.inc` in **Tools** > **Includes commands from files**.
   - Load `IIC_PRO.axf` from **Target** > **Load Image**…, and then run this `.axf` file.

2.1.6.2 Hardware Settings

To configure the i.MX37 3-Stack board, use the information in Table 2-5 to set the switches for the green Debug board.

Table 2-5 Switches on the i.MX37 3-Stack Board

Debug Board	
SW9	SW10
ON	ON

2.1.7 i.MX51 TO1 3-Stack Board

2.1.7.1 Prerequisites

Check that these prerequisites have been met:

1. Ensure that the chip is fused in "non-security" or "engineering" mode.
2. If the chip is not fused in "non-security" or "engineering" mode, use the `iim_hab_type_key_fuse_pgm.axf` and `Elvis_init.inc` files to fuse the chip to "engineering" mode, using these steps.
   — In RVDS, load `Elvis_init.inc` in **Tools** > **Includes commands from files**.
   — Load `iim_hab_type_key_fuse_prm.axf` from **Target** > **Load Image**…, and then run this `.axf` file.

2.1.7.2 Hardware Settings

To configure the i.MX51 TO1 3-Stack board, use the information in Table 2-7 to set the switches for the green Debug board. Others are set to OFF.

Table 2-7 Switches on the i.MX51 TO1 3-Stack Board

Debug Board		CPU version 2 Board	
SW9	SW10	SW2	
		Dip1	Dip2
ON	ON	ON	ON

2.1.8 i.MX51 TO2 3-Stack Board

To configure the i.MX51 TO2 3-Stack board, use the information in Table 2-8 to set the switches for the green Debug board. Others are set to OFF.

Table 2- 8 Switches on the i.MX51 TO2 3-Stack Board

Debug Board	
SW9	SW10
ON	ON

2.1.9 i.MX25 3-Stack Board

To configure the 1.MX25 3-Stack board, use the information in Table 2-9 to set the switches for the Debug board.

Table 2-9 Switches on the i.MX25 3-Stack Board

Debug Board	
SW9	SW10
ON	ON

2.2 Connecting the Host to the Board

Use the instructions in this section that pertain to your connector type.

2.2.1 USB Connection

To connect and use a USB, follow these steps:

1. Connect the USB cable to the USB port of the board.
2. Select of the following i.MX CPU that you will use.

   These versions support a USB connection:
   - i.MX27-TO1 for the i.MX27 TO1 board
   - i.MX27-TO2 for the i.MX27 TO2/i.MX27 3-Stack board
   - i.MX31_TO2 for the i.MX31 3-Stack board
   - i.MX32 for the i.MX32 ADS board
   - i.MX31_TO2.1 for the i.MX31 3-Stack board
   - i.MX35 for the i.MX35 3-Stack board

   NOTES

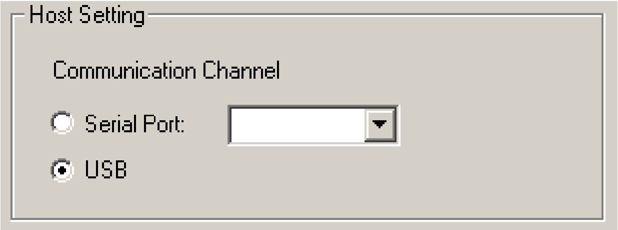
   If your PC displays the message “USB not recognized” when the USB is insert in the i.MX31 boards, we recommend that you use the “Switch UART to USB” method.

   i.MX31 TO1: ROM code does not support the USB download method.

   i.MX25: Downloading by USB is supported, but programming, dumping, and erasing by USB is not supported yet.

   i.MX37 prior to TO1.1.1: USB is not supported, due to a ROM code issue.

3. Power on the board and select USB for communications (Figure 2-1).


Host Setting

Communication Channel

○ Serial Port:

● USB

Figure 2-1 Selecting a USB Connection

2.2.2 UART Connection

To connect and use a UART, follow these steps:

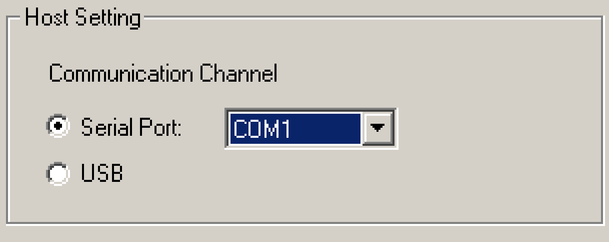
1. Connect a serial cable to the UART1 port of the ADS board.
2. Select the i.MX CPU that you will use.

   These versions support a UART connection:
   - i.MX27-TO1 for i.MX27 TO1 board
   - i.MX27-TO2 for i.MX27 TO2 board and i.MX27 3-Stack board
   - i.MX31-TO1 for i.MX31 TO1 ADS
   - i.MX31-TO2 for i.MX31 TO2 ADS board
   - i.MX31-TO2.1 for i.MX31 TO2.1 ADS board
   - i.MX32 for i.MX32 ADS board
   - i.MX35 TO1 for i.MX35 TO1 3-Stack board
   - i.MX35 TO2 for i.MX35 TO1 3-Stack board
   - i.MX37 for i.MX37 3-Stack board
   - i.MX51 TO1 for i.MX51 TO1 3-Stack board
   - i.MX51 TO2 for i.MX51 TO2 3-Stack board
   - i.MX25 for i.MX25 3-Stack board

   NOTE

   i.MX25: Set the UART baud rate to 57600, as limited by the ROM code. The low UART baud rate causes a reduced programming/dumping/erasing speed.

3. Power on the board.
4. Select the COM port that your PC will use (Figure 2-2).



Host Setting

Communication Channel

● Serial Port: COM1

○ USB

Figure 2-2 Selecting a UART Connection

2.2.3 Switching from UART to USB

NOTE

> i.MX31: This feature is available only for Flash programming, dumping, and erasing on the i.MX31 boards. It works only for USB ULPI PHY, not for USB Serial and Atlas USB.

To switch from UART to USB, use these steps:

1. Connect a serial cable to the UART1 port of the ADS board.

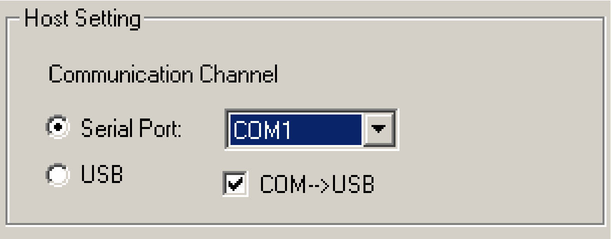
   Be prepared to use one USB cable, but do not plug the USB cable in yet.

2. Select the i.MX CPU:
   - i.MX31-TO1 for the i.MX31 TO1 ADS board
   - i.MX31-TO2 for the i.MX31 TO2 ADS and MX31 3-Stack boards

3. Power on the board.

4. Select the Serial Port and COM port that your PC will use.

5. Select **COM** > **USB** (Figure 2-3).



Host Setting

Communication Channel

● Serial Port: COM1

○ USB

☑ COM-->USB

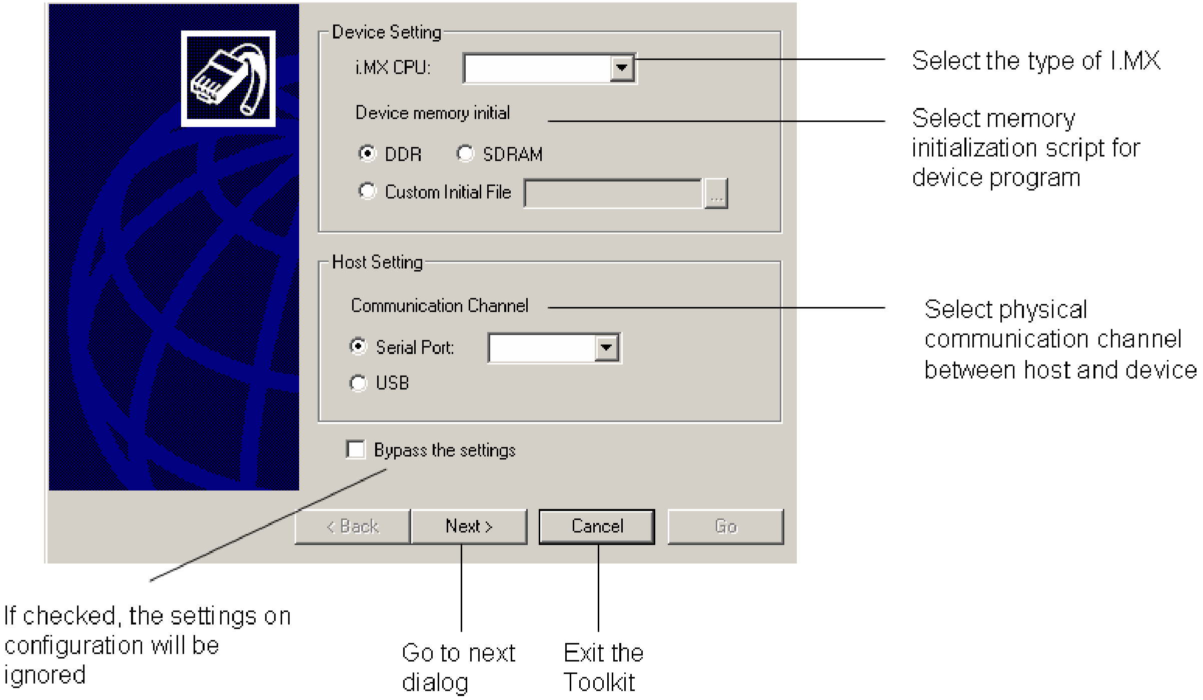
Figure 2-3 Switch UART to USB

To erase, dump, and program using the Flash Tool, plug in the USB cable using the instructions in Section 2.2.1, USB Connection.

.

Chapter 3
Configuring the ATK

When you launch the ATK, the configuration screen is displayed (Figure 3-1). For an example of a completed configuration, see Figure 3-2.



Select the type of I.MX

Select memory initialization script for device program

Select physical communication channel between host and device

If checked, the settings on configuration will be ignored

Go to next dialog

Exit the Toolkit

Figure 3-1 ATK Configuration Main Screen

To configure the ATK, use these steps:

1. In the Device Setting section, click the down arrow to select the i.MX CPU to use:
   - Select i.MX31-TO1 for the i.MX31 TO1 ADS board
   - Select i.MX31-TO2 for the i.MX31 TO2 board
   - Select i.MX31-TO2.1 for the i.MX31 TO2.1 board
   - Select i.MX32 for the i.MX32 ADS board
   - Select i.MX27-TO1 for the i.MX27 TO1 ADS board
   - Select i.MX27-TO2 for the i.MX27 TO2 board and i.MX27 3-Stack board
   - Select i.MX35 TO1 for the i.MX35 TO1 3-Stack board
   - Select i.MX35 TO2 for the i.MX35 TO2 3-Stack board
   - Select i.MX37 for the i.MX37 3-Stack board
   - Select i.MX51 TO1 for the i.MX51 TO1 3-Stack board
   - Select i.MX51 TO2 for the i.MX51 TO2 3-Stack board
   - Select i.MX25 for the i.MX25 3-Stack board.

NOTE

> The **i.MX CPU** option is mandatory even when **Bypass the settings** is selected.

2. For **Device memory initial** (**DDR** or **SDRAM**, **DDR2** or **MDDR**), select the memory initialization script to use for the device program.

   First, determine which DDR is used on your board:
   - **DDR**: Specifies the DDR initialization script.
   - **SDRAM**: Specifies the SDRAM initialization script.

   For the i.MX35 TO1/TO2 3-Stack board:
   - **DDR2**: Specifies the DDR2 initialization script.
   - **MDDR**: Specifies the MDDR initialization script.

   For the i.MX25 3-stack board:
   - **MDDR**: Specifies the MDDR initialization script.
   - **Custom Initial File**: If selected, click the button to select a custom memory initialization script. You can also use your own memory initialization file if it has the correct format. See the **Appendix** for details.

3. In the **Host Setting** section, select a physical communication channel between PC host and device:
   - **Serial Port**: Selects the UART as the communication channel.
   - **USB:** Selects the USB as the communication channel.
   - **COM** > **USB**: Switching from UART to USB **Communication Channel**. If **Serial Port** is checked, select a COM port (COM1 is preferred).

4. If **Serial Port** is selected and you want to switch to USB for dumping, erasing and programming, select **COM** > **USB**.

5. For **Bypass the settings**, use the following information:
   - — If selected, the settings on this configuration screen are ignored. Note that the ATK supports the tools independent of the configuration.
   - — If cleared, the tool will check the configuration settings, including opening the USB/COM port, and others.

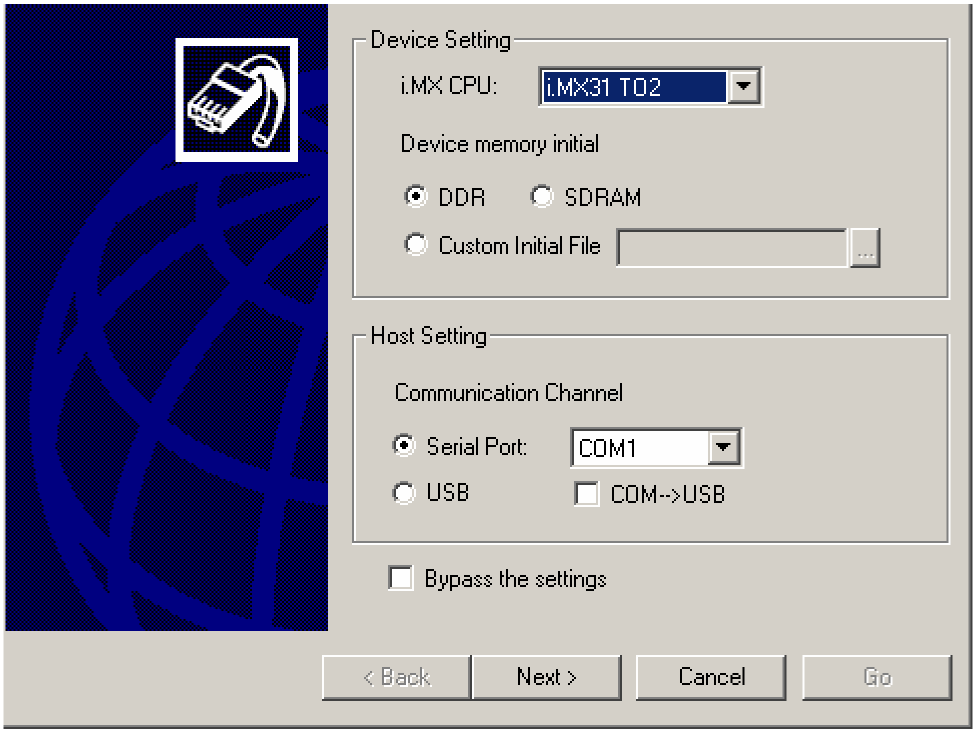
6. Click **Next.**

   The Select Advanced Tools dialog is displayed.

NOTE

> The Select Advanced Tools dialog will not be displayed if the board is not connected successfully, and if **Bypass the Settings** is not selected.

Figure 3-2 illustrates the settings for an ATK configuration for the i.MX31 TO2 with DDR memory initialization and a UART connection.



Device Setting

i.MX CPU: i.MX31 TO2

Device memory initial

● DDR ○ SDRAM

○ Custom Initial File

Host Setting

Communication Channel

● Serial Port: COM1

○ USB □ COM-->USB

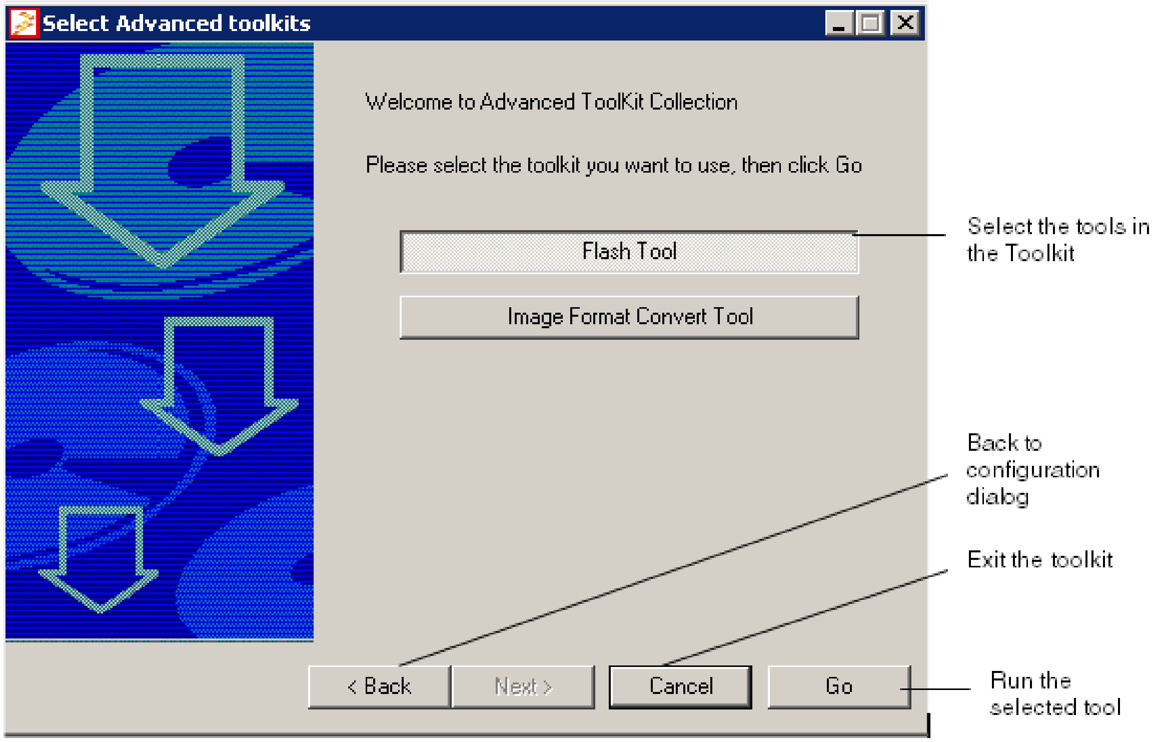
□ Bypass the settings

Figure 3-2 Sample Configuration

Chapter 4
Using the Flash Tool

You can use the Flash tool to download, program, dump, and erase images in Flash memory to the i.MX boards.

Figure 4-1 illustrates the Advanced ToolKit selection screen.



Select Advanced toolkits

Welcome to Advanced ToolKit Collection

Please select the toolkit you want to use, then click Go

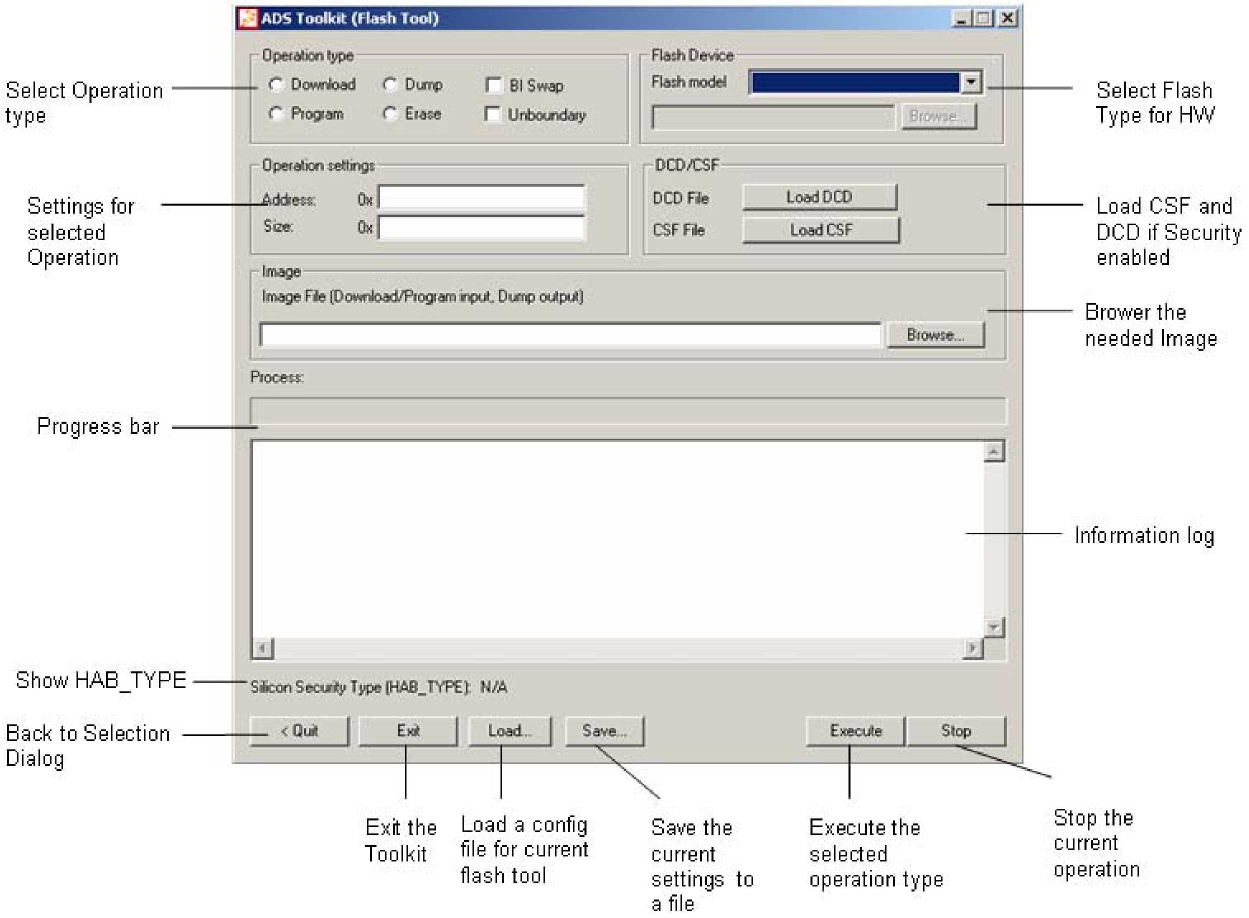
Flash Tool
Image Format Convert Tool
< Back
Next >
Cancel
Go
Select the tools in the Toolkit
Back to configuration dialog
Exit the toolkit
Run the selected tool

Figure 4-1 Selecting the Advanced Tools

To use the Flash tool, follow these steps:

1. In the Advanced Tools screen, click **Flash Tool**, and then click **Go.**

   The Flash Tool screen is displayed (Figure 4-2). Table 4-1 describes the Flash Tool options. For examples of the results of option selections, see Figure 4-3 through Figure 4-6.



Select Operation type

Settings for selected Operation

Progress bar

Show HAB_TYPE

Back to Selection Dialog

Select Flash Type for HW

Load CSF and DCD if Security enabled

Brower the needed Image

Information log

Exit the Toolkit

Load a config file for current flash tool

Save the current settings to a file

Execute the selected operation type

Stop the current operation

Figure 4-2 Flash Tool

2. In the **Operation Type** section, select the operation to perform.

   Generally, you should select **BBT** for Linux images with a Flash Model type of NAND, and **Bi Swap** for Windows Embedded CE images with a Flash Model type of NAND and a size exceeding 2k per page.

   For certain Windows Embedded CE image cases, refer to the User’s Guide to determine whether **Bi Swap** is implemented in the software. If it is not implemented, **Bi Swap** should be cleared. For more information, see Section 6.3 "Why and How to Check BBT| BI Swap in the Flash Tool".

3. In the Flash Model field, select the type (this option is unavailable if you selected **Download** as the **Operation Type**). These are the supported Flash memories for specific boards:

For i.MX31 ADS boards, the supported Flash memories are:

NOR (Spansion)	Address from: 0xA0000000
NAND	Address from: 0x00000000

For i.MX31 3-Stack boards, the supported Flash memory is:

NAND	Address from: 0x00000000
NOR (Spansion)	Address from: 0xA0000000
SD	Address from: 0x00000000
MMC	Address from: 0x00000000

For i.MX32 3-Stack boards, the supported Flash memory is:

NOR (Spansion)	Address from: 0xA0000000
NAND	Address from: 0x00000000
SD	Address from: 0x00000000
MMC	Address from: 0x00000000

For i.MX35 TO1/TO2 3-Stack boards, the supported Flash memories are:

NOR (Spansion)	Address from: 0xA0000000
NAND	Address from: 0x00000000

For i.MX37 3-Stack boards, the supported Flash memory is:

NAND	Address from: 0x00000000

For i.MX51 TO1/TO2 3-Stack boards, the supported Flash memory is:

NAND	Address from: 0x00000000

For i.MX25 3-Stack boards, the supported Flash memory is:

NAND	Address from: 0x00000000

NOTE

If you have built your own library for the target board, you can specify a different Flash Model.

You do not need to know the specific type of NAND or NOR Spansion flash in use.

4. Enter the address where the operation should start, and the size (which is entered only for dumping and erasing operations).

5. Browse to the file to be programmed, downloaded, dumped, or erased.

6. Select **program**/**download**/**dump**/**erase** to start the operation, and wait until it is complete.

   Error information is logged.

NOTE

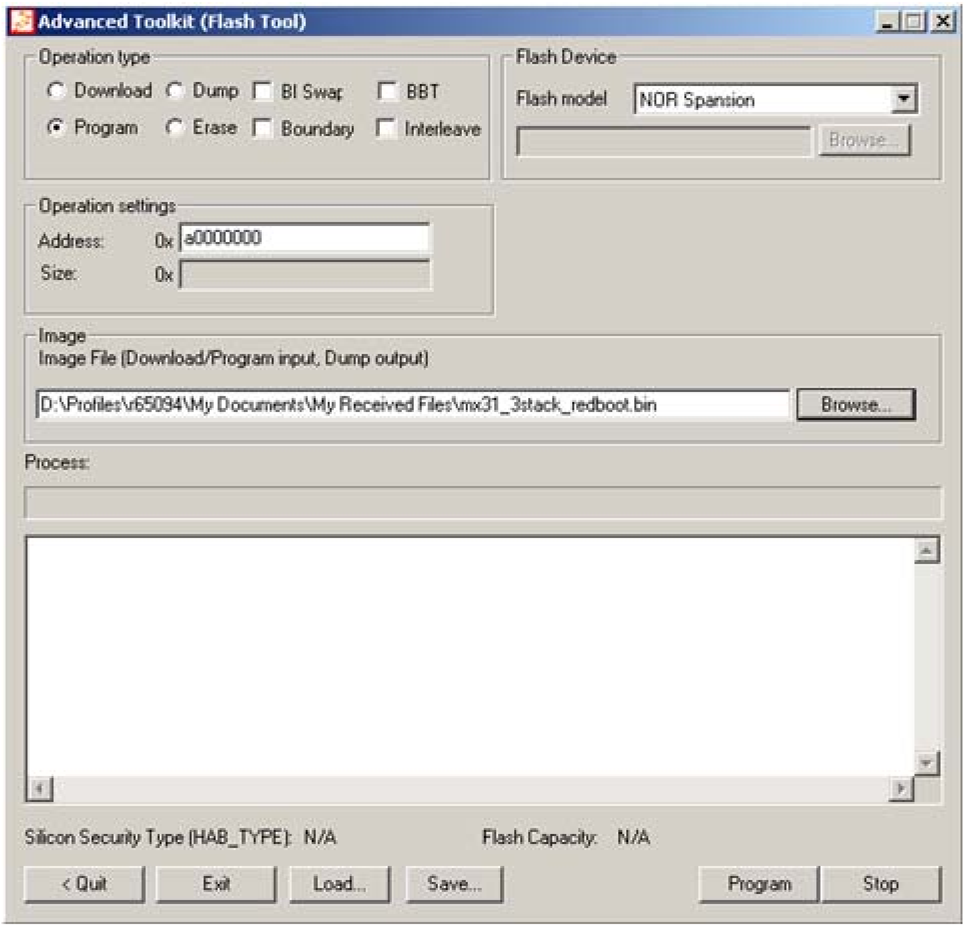
If **COM** > **USB** is selected in configuration, follow any instructions that appear when performing any operations.

Table 4-1 Flash Tool Options

Section	Option	Description
Operation Type	Download	Downloads the image to the address specified, and then executes directly at the downloaded address.
	Dump	Dumps a selected area of Flash device to a file.
	Program	Programs a selected area of Flash from a file.
	Erase	Erases a selected area of Flash.
	BI Swap	Swaps Bad block identifier data, which is one solution for Bad block identifier data in the case that Nand Flash is incompatible with Nand Flash Controller.
	BBT	Bad block Table, which is the table recording all bad block information.
	Unboundary	The Flash tool supports two types of Flash operations: (1) Programs the Flash with block alignment. If the programming size is less than the block size, the current block is erased and is written with the programming data in the header. The original data in the block is lost. (2) Programs the Flash with byte alignment. That means just programming the Flash with the input data size. If the programming size is less than the block size, the remaining data in the block will be kept the same as the original. If checked, then type (2) operation is used.
	Interleave **i.MX51 only**	An i.MX51 board with multiple NAND flash chips (for example, 4 chips) has two access modes: (1) Interleave mode indicates parallel access to the NAND flashes. The data is distributed equally into the multiple chips. For example, if the board has 4 chips, each chip will be 2K page size. Writing 8K data to NAND flash in interleave mode occurs in the following sequence: the first 2K goes to chip1, the second 2K to chip 2, the third 2K to chip 3, and last 2K goes to chip 4. Interleave mode programming enables the multiple chips to be programmed at the same time, thus improving write performance. (2) Non-interleave mode indicates exclusive access to the NAND flashes. In this mode, only one of the multiple chips on the board can be accessed at a time. The programming data will go to the selected NAND flash.
Flash Device	Flash Model	Specifies the Flash type for the associated board when the Program, Dump, or Erase Flash operations are selected: NOR (SPansion) for the i.MX31/i.MX27/i.MX32/i.MX35 boards NAND for the i.MX31//MX27/i.MX32/i.MX35/i.MX37/i.MX51/i.MX25 board MMC for the i.MX31/i.MX32/i.MX35 /i.MX37/i.MX51/i.MX25 board SD for the i.MX31/i.MX32/i.MX35/i.MX37/i.MX51/i.Mx25 boards USER INPUT for a user input library
	Flash Lib (available when USER INPUT is selected)	Device program running on the board that handles the communication with the PC host through the protocol to program/dump/erase the flash.
Operation Settings	Address	Address in the physical memory of the target board where the operation will start.

Section	Option	Description
	Size	Size in bytes for erase and dump operations.
Image	Image file	For downloading operations, indicates the image to be downloaded to RAM. For Flash operations, indicates the image to be downloaded to Flash. For dumping operations, indicates the file that is used to store the dumped content.
Process	Progress bar	Displays the completed percent of the selected operation.
Information Log		Displays the execution log.

Figure 4-3 displays options for programming an image to 0xA0000000 in NOR flash.



○ Download ○ Dump □ BI Swap □ BBT

● Program ○ Erase □ Boundary □ Interleave

Flash model: NOR Spansion

Address: 0x a0000000

Size: 0x

Image File (Download/Program input, Dump output)

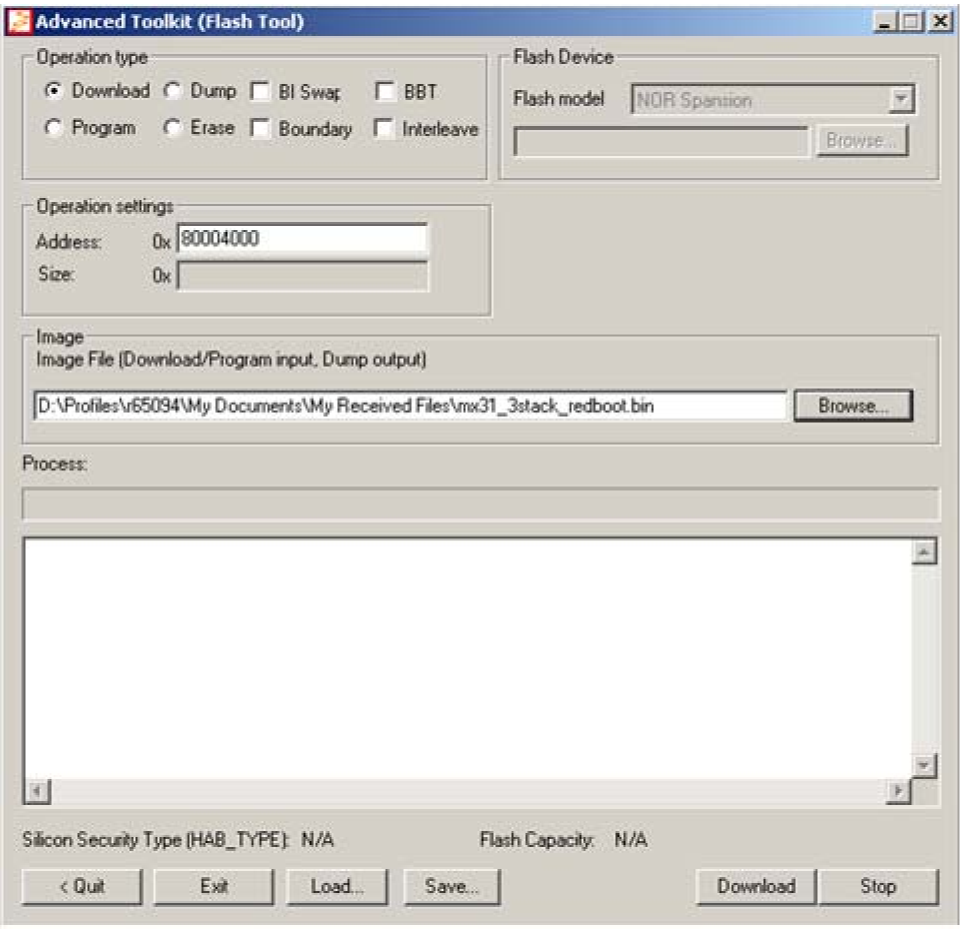
D:\Profiles\r65094\My Documents\My Received Files\mx31_3stack_redboot.bin

Silicon Security Type (HAB_TYPE): N/A

Flash Capacity: N/A

Figure 4-3 Programming NOR Flash

Figure 4-4 displays options for downloading an image to 0x80004000 in RAM.



Advanced Toolkit (Flash Tool)

Operation type

● Download ○ Dump □ BI Swap □ BBT

○ Program ○ Erase □ Boundary □ Interleave

Flash Device

Flash model: NOR Spansion

Operation settings

Address: 0x 80004000

Size: 0x

Image

Image File (Download/Program input, Dump output)

D:\Profiles\r65094\My Documents\My Received Files\mx31_3stack_redboot.bin

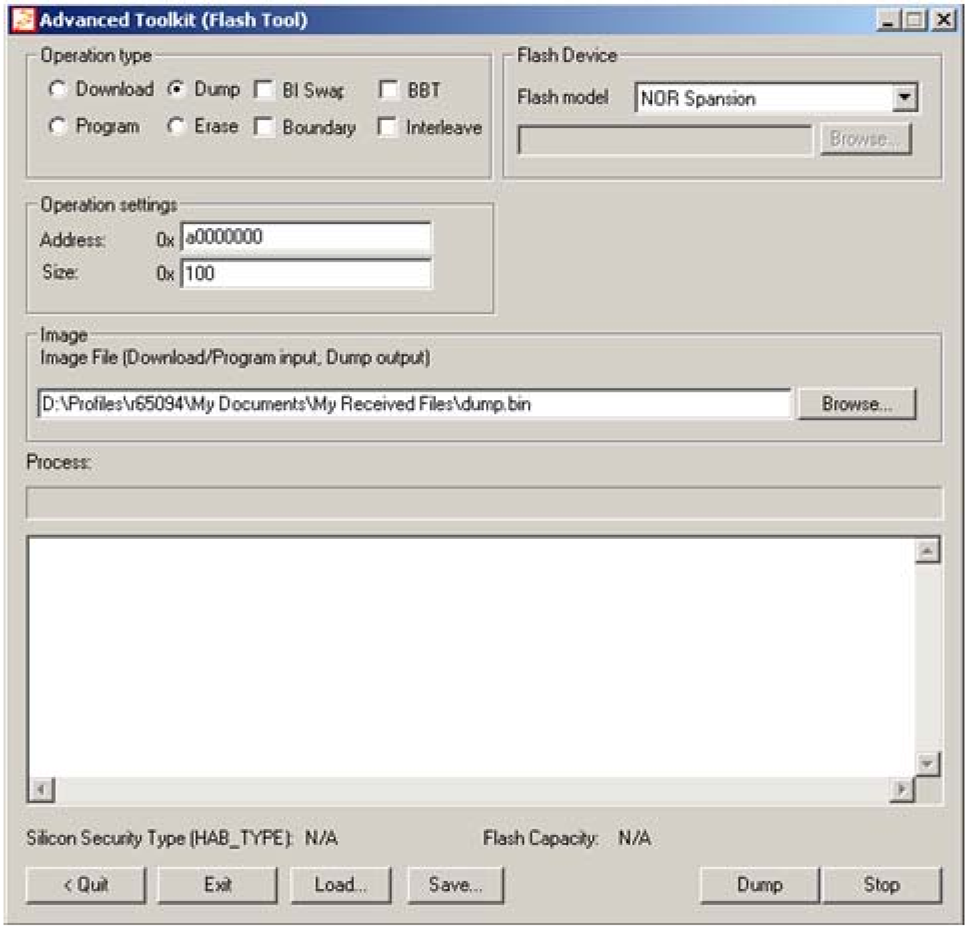
Process:

Silicon Security Type (HAB_TYPE): N/A

Flash Capacity: N/A

Figure 4-4 Downloading an Image to RAM

Figure 4-5 displays options for using the Flash Tool to dump from 0xA0000000 in NOR Flash with size 0x100.



Advanced Toolkit (Flash Tool)

Operation type

○ Download ● Dump □ BI Swap □ BBT

○ Program ○ Erase □ Boundary □ Interleave

Flash Device

Flash model: NOR Spansion

Operation settings

Address: 0x a0000000

Size: 0x 100

Image

Image File (Download/Program input, Dump output)

D:\Profiles\r65094\My Documents\My Received Files\dump.bin

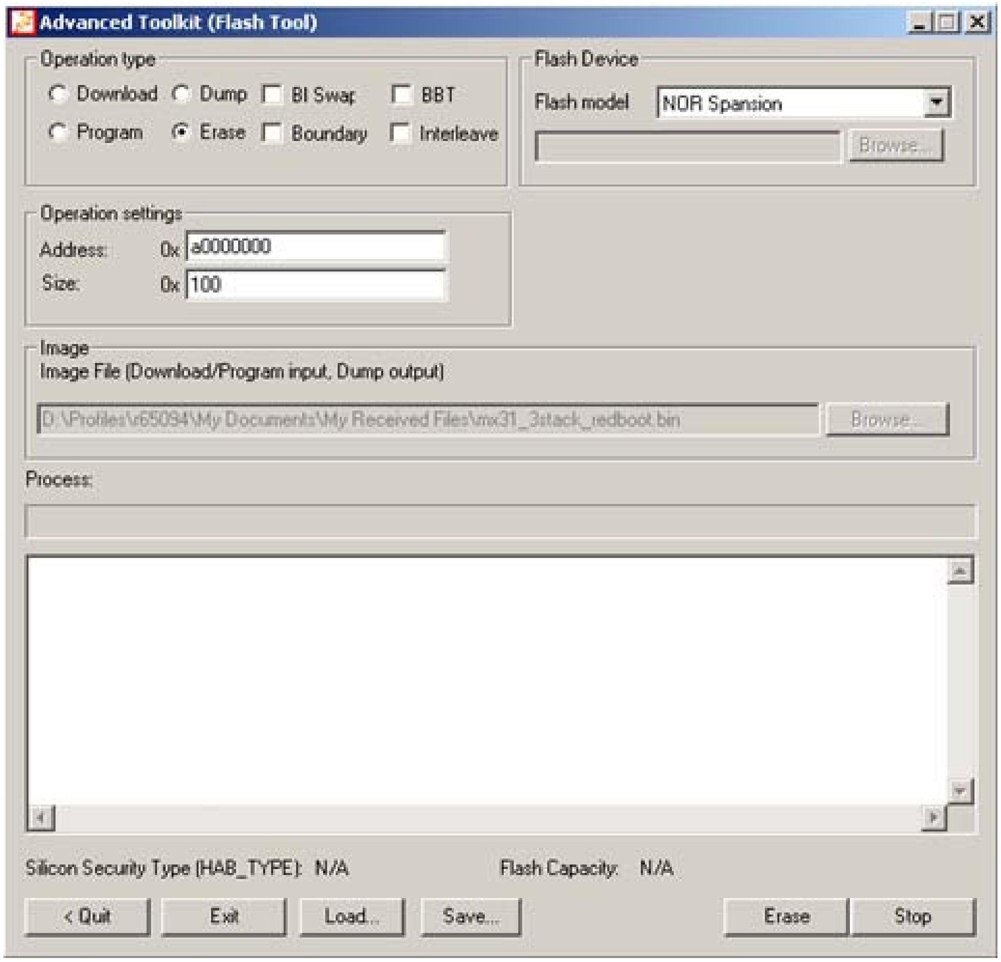
Process:

Silicon Security Type (HAB_TYPE): N/A

Flash Capacity: N/A

Figure 4-5 Dumping from NOR Flash

Figure 4-6 displays options for using the Flash Tool to erase 0x100 bytes from 0xa0000000 in NOR Flash.



Advanced Toolkit (Flash Tool)

Operation type

○ Download ○ Dump □ BI Swap □ BBT

○ Program ● Erase □ Boundary □ Interleave

Flash Device

Flash model: NOR Spansion

Operation settings

Address: 0x a0000000

Size: 0x 100

Image

Image File (Download/Program input, Dump output)

D:\Profiles\r65094\My Documents\My Received Files\mx31_3stack_redboot.bin

Process:

Silicon Security Type (HAB_TYPE): N/A

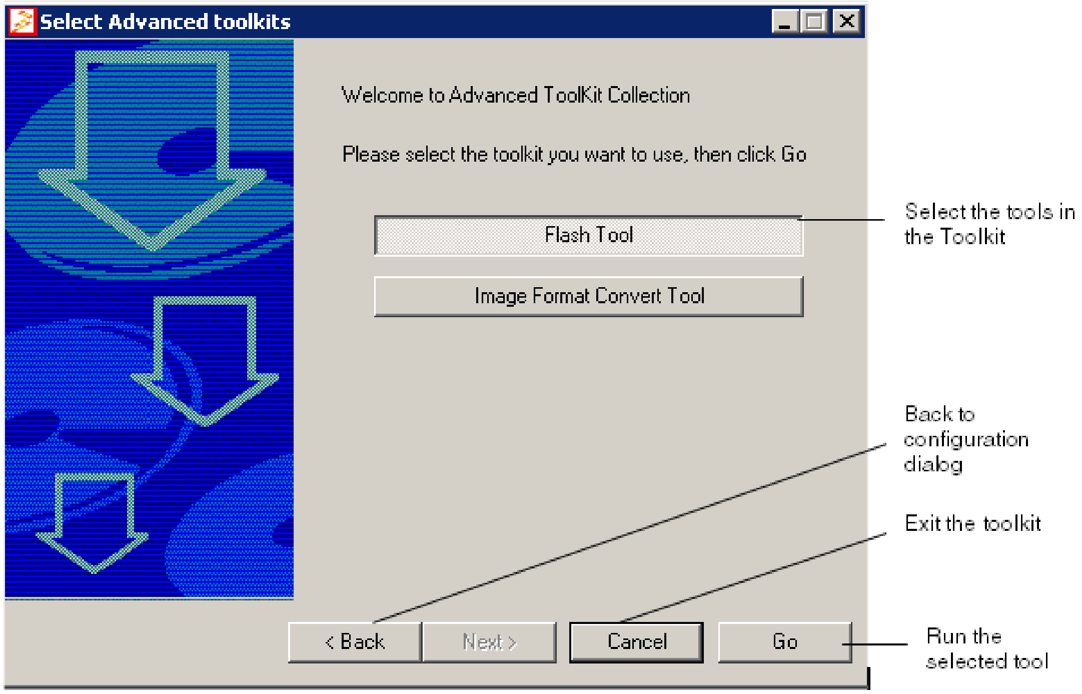
Flash Capacity: N/A

Figure 4-6 Erasing in NOR Flash

Chapter 5
Converting Image Formats

You can use the Image Format Convert Tool to convert the following image file formats: binary to S record, S-record to binary, and ELF to binary.

Figure 5-1 displays the Advanced ToolKit selection screen.



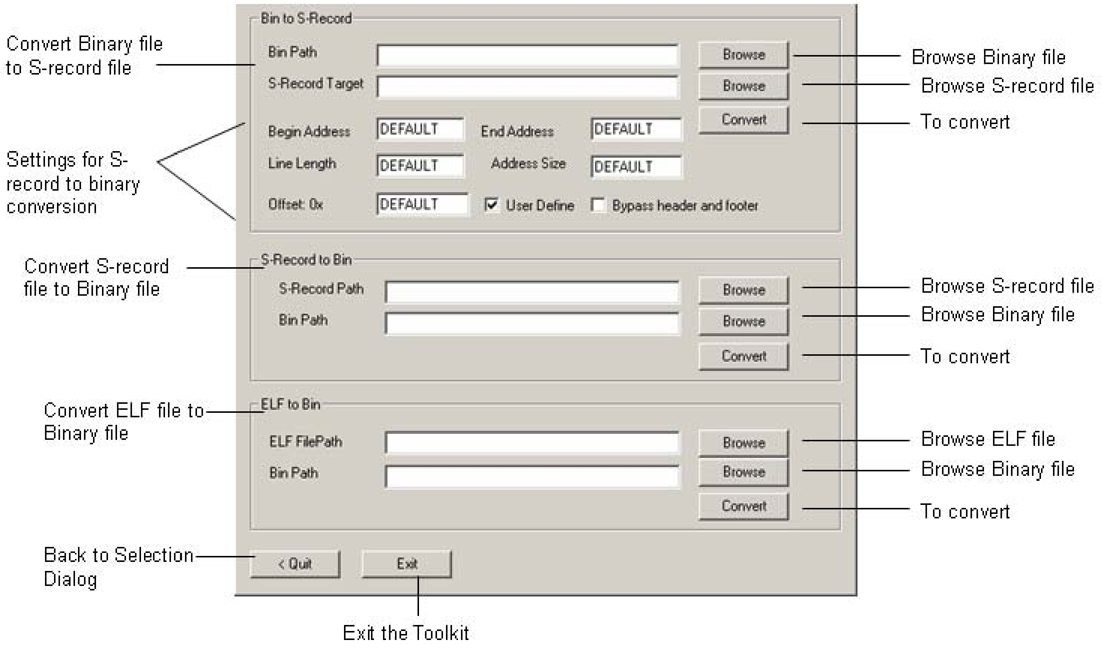
Select Advanced toolkits
Welcome to Advanced ToolKit Collection
Please select the toolkit you want to use, then click Go
Flash Tool
Image Format Convert Tool
< Back
Next >
Cancel
Go
Select the tools in the Toolkit
Back to configuration dialog
Exit the toolkit
Run the selected tool

Figure 5-1 Selecting Advanced Tools

To use the Image Convert tool, follow these steps:

1. In the Advanced Tools screen, click **Image Format Convert Tool**, and then click **Go**.

   The Image Convert tool screen is displayed (Figure 5-2).
   - For a detailed description of the options, see Table 5-4, which follows these instructions.
   - For examples of option choices, see Figure 5-3, Figure 5-4, and Figure 5-5.



Begin Address DEFAULT End Address DEFAULT
Line Length DEFAULT Address Size DEFAULT
Offset: 0x DEFAULT ☑ User Define ☐ Bypass header and footer

Convert Binary file to S-record file

Settings for S-record to binary conversion

Convert S-record file to Binary file

Convert ELF file to Binary file

Back to Selection Dialog

Browse Binary file

Browse S-record file

To convert

Browse S-record file

Browse Binary file

To convert

Browse ELF file

Browse Binary file

To convert

Exit the Toolkit

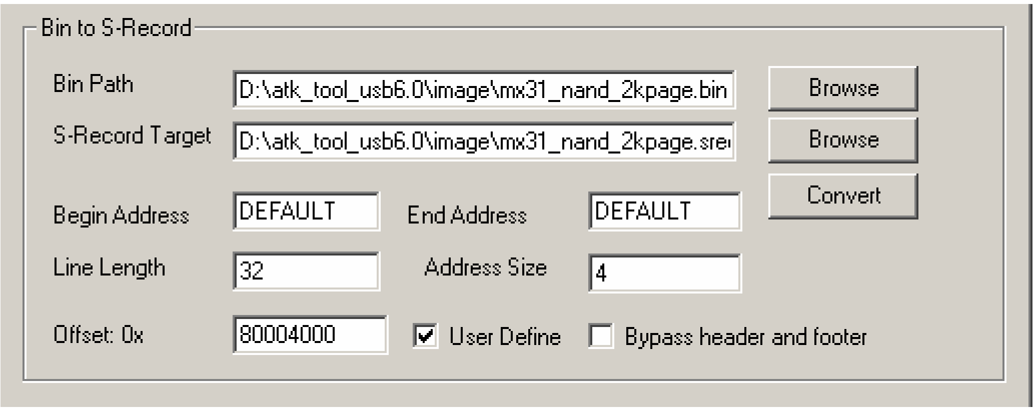
Figure 5-2 Image Conversion Tool

2. Select a file type to convert, enter the required information, and then click **Convert**.
   - For converting a **binary formatted file** to an **S record format**, enter the location of the image file, the location for the converted file, address information, line length, address size, and offset parameters, and then click **Convert**.
   - For converting an **S-record formatted file** to a **binary file format**, enter the location of the file to convert and the location for the converted file, and then click **Convert**.
   - For converting an **ELF formatted file** to a **binary file format**, enter the location of the file to convert and the location for the converted file, and then click **Convert**.

Table 5-1 Image Convert Tool Parameters

Item	Sub-Item	Description
Bin to S-record	Bin Path	Selects the binary file to be converted.
	S-record Target	Selects the location for the output S-record file.
	Begin Address	The beginning address of the file to be converted. The default address is 0, which means that the conversion starts from the beginning of the file.
	End Address	The end address of the file to be converted. The default End Address is the length of the file.
	Line Length	The S-record line length; the range is 8 – 32. By default, the tool calculates the length according to the address and the offset.
	Address Size	The address length in bytes; the range is 2 - 4. By default, the tool calculates the size according to the address and the offset.
	Offset	The offset of the part to be converted in the memory layout. The offset plus the beginning address become the first address in the memory layout.
	User Define	If selected, you can enter the info; if not, the tool calculates the values for you.
	Bypass header and footer	If selected, the tool will not generate the header and footer.
	Convert	Converts the binary file to the S-record file.
S-Record to Bin	S-Record Path	Selects the s-record file to be converted.
	Bin Path	Selects the output path where the output binary file will be put.
	Convert	Converts the S-record file to a binary file.
ELF to Bin	ELF File Path	Selects the ELF file to be converted.
	Bin Path	Selects the output path where the output binary file will be put.
	Convert	Converts the ELF file to a binary file.

Figure 5-3 illustrates converting a binary formatted file to an S record format.



Bin to S-Record

Bin Path: D:\atk_tool_usb6.0\image\mx31_nand_2kpage.bin

S-Record Target: D:\atk_tool_usb6.0\image\mx31_nand_2kpage.sre

Begin Address: DEFAULT

End Address: DEFAULT

Line Length: 32

Address Size: 4

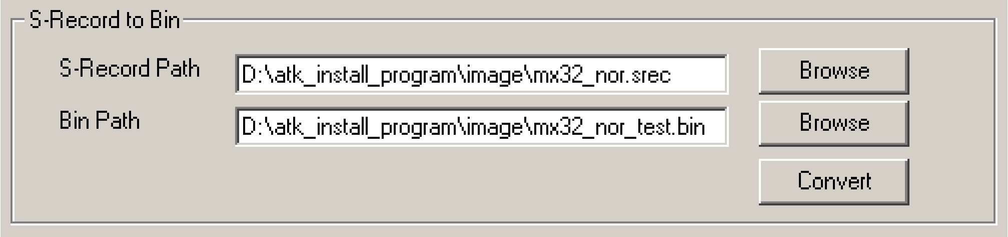
Offset: 0x 80004000

☑ User Define

☐ Bypass header and footer

Figure 5-3 Binary to S Record Example

Figure 5-4 illustrates converting an S-record formatted file to a binary file format.



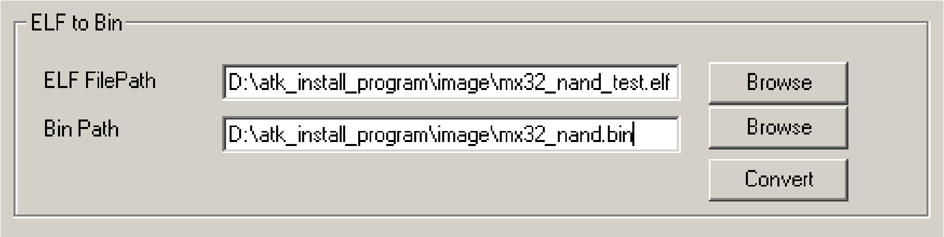
S-Record to Bin

S-Record Path: D:\atk_install_program\image\mx32_nor.srec

Bin Path: D:\atk_install_program\image\mx32_nor_test.bin

Figure 5-4 S Record to Binary Example

Figure 5-5 illustrates converting an ELF formatted file to a binary file format.



ELF to Bin

ELF FilePath: D:\atk_install_program\image\mx32_nand_test.elf

Bin Path: D:\atk_install_program\image\mx32_nand.bin

Figure 5-5 ELF to Binary Example

Chapter 6
Additional Information

This chapter provides information about the memory initialization file format, configuring the ATK for high-speed USB on the i.MX27 TO2, and the purpose and method for checking the BBT/BI swap in the flash tool.

6.1 Memory Initialization File Format

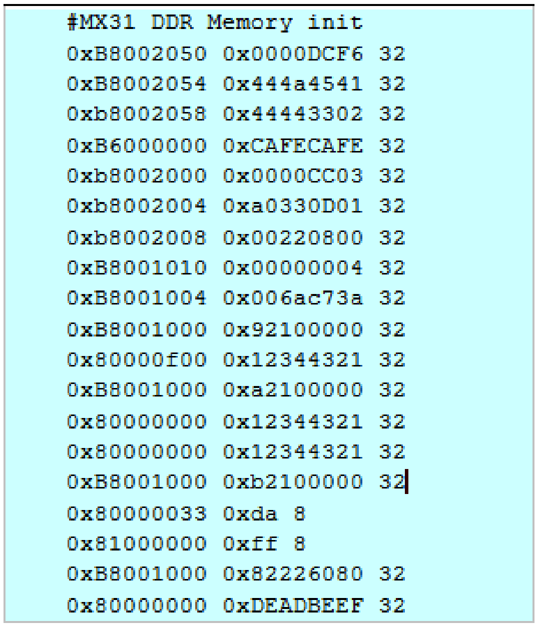
The memory initialization file is a text (.txt) file in which each line contains a Register Write operation to the MX31 device. The file columns contain information in the following format: **Address Data Format**, where:

Address (hexadecimal)	The register address to be written to; it must be in hexadecimal and begin with 0x.
Data (hexadecimal)	The value written to the register address; it must be in hexadecimal and begin with 0x
Format (decimal)	Indicates the data access type: 8-bit (8), 16-bit (16) or 32-bit (32).

As seen in Figure 6-1, the Address/Data/Format items are separated by a single space. End each line by pressing **Enter**, and then start on the next line. Use '#' to begin a comment line. The following example file is located in:

`<tool_install_path>\example\memory_init\mx31_ddr_init.txt`

In the first line of the example file image, the columns contain the following information: 0xB8002050 is the address; 0x0000DCF6 is the data value; and 32 indicates the format.



```
#MX31 DDR Memory init
0xB8002050 0x0000DCF6 32
0xB8002054 0x444a4541 32
0xb8002058 0x44443302 32
0xB6000000 0xCAFECAFE 32
0xb8002000 0x0000CC03 32
0xb8002004 0xa0330D01 32
0xb8002008 0x00220800 32
0xB8001010 0x00000004 32
0xB8001004 0x006ac73a 32
0xB8001000 0x92100000 32
0x80000f00 0x12344321 32
0xB8001000 0xa2100000 32
0x80000000 0x12344321 32
0x80000000 0x12344321 32
0xB8001000 0xb2100000 32
0x80000033 0xda 8
0x81000000 0xff 8
0xB8001000 0x82226080 32
0x80000000 0xDEADBEEF 32
```

Figure 6-1 Example File Image

6.2 Configuring ATK for High-Speed USB on i.MX27 TO2

To support the i.MX27 TO2 high-speed USB boot, you must modify HW/CPLD.

Use these steps:

1. Remove RP11 and RP12.
2. Remove R240.
3. Set "0" to bit 7 in CPLD_BCTRL3.

Where:

Bit 7 (OTG_HS_EN): USB OTG **High Speed Enable** enables the USB OTH High Speed interface on the CPU.

0 = OTG High Speed Interface enabled

1 = OTG High Speed Interface disabled

6.3 Why and How to Check the BBT| BI Swap in the Flash Tool

Why do we need the BI swap? The FSL NFC memory layout differs from the NAND flash data layout when it comes from the factory. The layout will cause the BI flag of the NAND flash to fail in the main area of the NFC memory. This means that the BI flag will be overwritten by the real data, which will cause the BI flag to be lost. In order to keep the BI flag, you must swap the data corresponding to the BI flag position to an unused spare area position when writing NAND flash, and then swap it back when it is read.

There are two types of bad block management schema:

a) **Scan mechanism**: In order to detect whether the block is bad, the software must scan the block to test the BI Flag value. If none are specified as 0xff, it is a bad block.

b) **Flash-based bad block table**: The bad block information for all blocks is stored in the NAND flash after the first scan out. The bad block table will be used to determine if the block is bad.

Setting the BBT/Bi swap flag in the ATK:

— **Windows** bad block management schema: If you are using the nb0 file format, which is the Windows Embedded CE image, select the Bi swap flag for the program/dump/erase operation. Do NOT select the BBT flag.

— **Linux** bad block management schema: Select the BBT flags for program/dump/erase operation. If the image that you programmed also uses the BI swap to maintain the NAND flash BI flag, then also select the BI swap. If you are not sure, contact the image provider.

If you erased the bad block table on the NAND flash:

If you erased the partition that contains the BBT and the BBT flag was set, the ATK will reconstruct the BBT and store it into the NAND flash through the scan mechanism.

If you erased the partition that contains the BBT and the BBT flag was NOT set, the ATK will help you reconstruct the BBT and store it into NAND flash through the scan mechanism the next time you set the BBT flag for program/dump/erase operation.

NOTE

For Linux users, your work will be easier if you always select the BBT flags. If you not sure of the source or contents of the NAND flash, erase the entire NAND flash first, and then perform the other operations.