

---

# **i.MX35 SDK**

# **Windows Embedded CE 6.0**

# **Multimedia Framework**

## **User's Guide**

**Document Number: 08-7254-USRGD-ZCH66**

**Rev. 1.7**

**12/2009**

## ***How to Reach Us:***

### **Home Page:**

[www.freescale.com](http://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](http://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](http://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](mailto: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](mailto: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](mailto: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.

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

---

# Contents

<b>About This Book .....</b>	<b>V</b>
Audience .....	v
Organization.....	v
Conventions .....	v
References.....	v
 <b>Chapter 1 Installation.....</b>	 <b>1</b>
1.1 Installing the BSP.....	1
1.2 Installing the SDK.....	2
1.2.1 Checking the Installation .....	5
1.2.1.1 Check the SDK folder in the BSP code tree .....	5
1.2.1.2 Check the BSP modification performed by the installer.....	6
1.2.1.3 Check the workspace modified by the installer .....	6
1.2.2 Removing an Existing Installed SDK .....	7
 <b>Chapter 2 Building OS Images .....</b>	 <b>8</b>
2.1 Working with Multimedia Components .....	8
2.1.1 Importing the Windows Embedded CE 6.0 Catalog.....	8
2.1.2 Add/Remove components from Catalog.....	9
2.2 Building OS Images .....	9
2.2.1 Building an Image in Platform Builder.....	9
2.2.2 Building an Image using the Command Line.....	9
2.3 Building Multimedia Components in sample Workspace .....	10
2.3.1 Building all components simultaneously .....	10
2.3.2 Building individual components .....	10
 <b>Chapter 3 Using the Test Procedures.....</b>	 <b>11</b>
3.1 Audio Decoder Test .....	11
3.2 Video Decoder Test .....	11
3.3 Image Decoder Test .....	11
3.4 BSAC Decoder Test.....	12
3.5 MP3 Encoder Test.....	12

3.6	SBC Encoder Test.....	13
3.7	WMA8 Encoder Test .....	14
3.8	PEQ Audio Post-processing Test.....	15
3.9	G.726 Encoder/Decoder Test.....	17
3.10	JPEG Encoder Test .....	18
3.11	S/PDIF Transfer Test .....	19
3.12	S/PDIF Receive Test.....	20
3.13	Deinterlacer Enable/Disable .....	21
3.14	DownMix Audio Post-processing Test.....	22
3.15	ASRC Post-processing Test.....	22
<b>Appendix A Patches and Fixes.....</b>		<b>I</b>
A.1	Unaligned Access Fault Disable .....	I
A.2	Video Memory Cache Policy .....	I
<b>Appendix B WinCE Kernel Debugger Configuration.....</b>		<b>II</b>

---

## About This Book

This guide explains how to install, build, and execute Windows Embedded CE 6.0 OS images for the 3-Stack board, using the Freescale i.MX35 3-Stack Windows CE Multimedia Framework Software Development Kit (SDK).

## 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      Explains how to install/uninstall the SDK.
- Chapter 2      Explains how to build Windows CE OS images using the BSP.
- Chapter 3      Explains how to test the multimedia components that you install.

## Conventions

This document uses the following conventions:

<i>Courier</i>	Is used to identify commands, explicit command parameters, code examples, expressions, data types, and directives.
<i>Italic</i>	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.MX35 PDK 1.7 Hardware User's Guide
2. i.MX Advanced Toolkit Standard Version User's Guide
3. Windows Embedded CE 6.0 BSP for i.MX35 3-Stack User's Guide



---

# Chapter 1

## Installation

The Freescale Multimedia Framework Software Development Kit (SDK) is a collection of binary, code, and support files that you can use to create Windows CE OS images for the i.MX35 3-Stack board. The SDK is distributed as a single archive EXE (.exe) file.

There are two steps to perform for the installation: first, install the i.MX35 3-Stack Board Support Package (BSP), and then install the SDK into the Windows CE source code tree and the Platform Builder development environment.

### 1.1 Installing the BSP

To install the BSP, use these steps:

1. Confirm that the BSP version is PDK1.7 version or above.
2. Follow the steps in Chapter 1 of the *Windows Embedded CE 6.0 BSP for i.MX35 3-Stack User's Guide*.

## 1.2 Installing the SDK

To install the SDK, use these steps:

### NOTE

Before installing the SDK, save any modified files and close the sample workspace, because the Installer will modify the file of the sample workspace project.

1. Run the installation execution package.

The Setup Wizard screen is displayed (Figure 1-1).

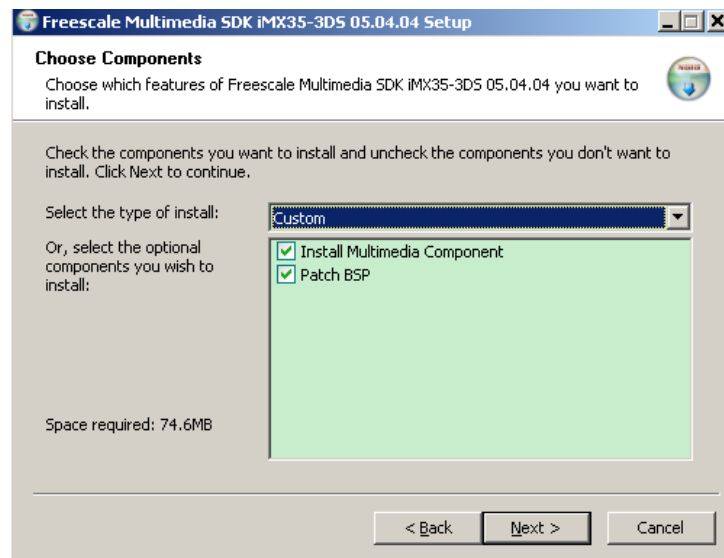


Figure 1-1 Setup Screen

2. Click Next

The Choose Components screen is displayed (Figure 1-2).

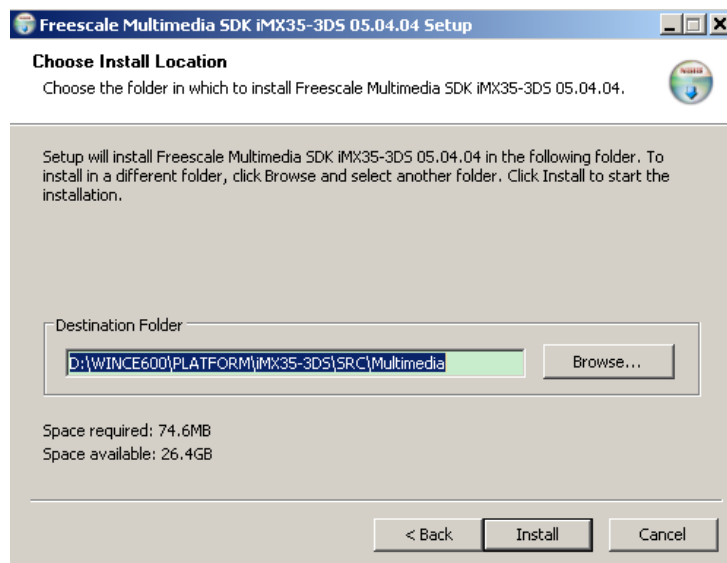




**Figure 1-2 Choose Components**

3. Click **Next**.

The installation location screen is displayed.



**Figure 1-3 Selecting the Installation Location**

4. Under Destination Folder, set the path of the destination folder for the SDK installation. By default, the multimedia framework SDK will be installed in the following path:  
`"$(WINCEROOT)\Platform\<platform_name>\src\Multimedia"`  
 where:  
 — `$(WINCEROOT)` is the path of your Windows CE root folder – "WINCE600"

- *<platform\_name>* is the name of the 3-stack platform BSP directory – “iMX35-3DS-PDK1\_7” for i.MX35 3-stack Windows Embedded CE 6.0 BSP
5. Continue with the installation wizard until the installation is complete.
  6. In the final installation window (Figure 1-4) you can select whether to do a clean build for the Windows CE OS Image automatically.
    - If **Make Run-time Image (Nk.bin)** is selected, the entire image build procedure, which includes the CSP build, BSP build, and SYSGEN, will be implemented automatically.
    - During the automatic build, all of the FSL developed multimedia components are included in the OS Image by default.

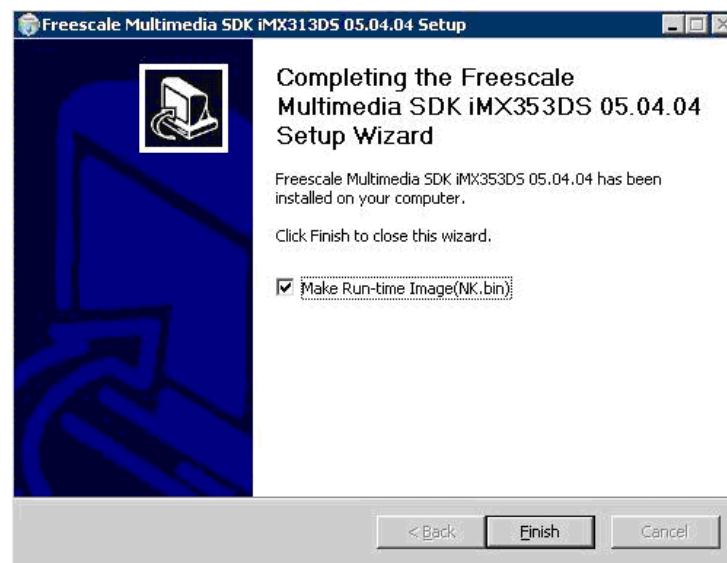


Figure 1-4 Completing the Installation

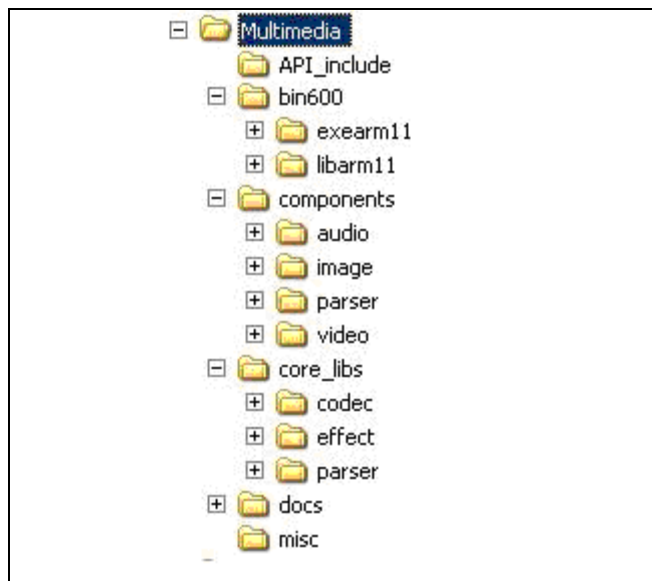
## 1.2.1 Checking the Installation

This section explains how to ensure that the multimedia framework SDK was installed successfully. The installer copies the Multimedia SDK folder into the BSP code tree:

- For Windows Embedded CE 6.0, the folder is copied to  
`\WINCE600\platform\<platform>\src`

### 1.2.1.1 Check the SDK folder in the BSP code tree

The SDK code tree architecture is shown in Figure 1-5. The table that follows describes the folder contents.



**Figure 1-5 SDK Code Tree**

Folder	Contents
API_include	This folder contains the global header files of multimedia DirectShow DMO and Filters.
components	This folder, with its audio, video, parser, and image subfolders , contains the DirectShow DMO and Filters source code of multimedia components.
core_libs	This folder contains header files and library binaries of multimedia components.
Docs	This folder contains API document, Datasheet and release notes of all of the components.
Misc	This folder contains the *.reg and *.bib file required by platform builder.
bin600	These folders contain DLL and EXE binaries, which are built by the

	components' source code.
--	--------------------------

### 1.2.1.2 Check the BSP modification performed by the installer

The SDK installer modifies the BSP to support the Freescale multimedia components.

To check the BSP modifications, use these steps:

1. In the **bsp\_cfg.h** file under `$( _WINCEROOT ) \platform \<platform> \src \inc \`, check that the line **BSP\_OAL\_DISABLE\_ALIGNMENT\_FAULT** definition has been uncommented as follows:

```
#define BSP_OAL_DISABLE_ALIGNMENT_FAULT
```

2. In the same **bsp\_cfg.h** file, check that the line **BSP\_VID\_MEM\_CACHE\_WRITETHROUGH** has been defined as **TRUE**, as follows:

```
#define BSP_VID_MEM_CACHE_WRITETHROUGH TRUE
```

3. In the **platform.reg** file under `$( _WINCEROOT ) \platform \<platform> \files \`, check the following line was added:

```
#include "$(_TARGETPLATROOT) \src \Multimedia \misc \fslmm_mx35.reg"
```

4. In the **platform.bib** file under `$( _WINCEROOT ) \platform \<platform> \files \`, check that the following line was added:

```
#include "$(_TARGETPLATROOT) \src \Multimedia \misc \fslmm_mx35.bib"
```

5. In the **directx.bib** file under `$( _WINCEROOT ) \public \directx \oak \files \`, check that the following line as below has been commented as follows:

```
;wmadmod.dll      $(_FLATRELEASEDIR) \wmadmod.dll      NK      SH
```

### 1.2.1.3 Check the workspace modified by the installer

The SDK installer modifies the Platform Builder's workspace to support the Freescale multimedia components.

To check the workspace, use these steps:

1. In Platform Builder IDE, select menu **Project -> iMX35-3DS-Mobility-PDK1\_7 Properties -> Configuration Properties -> Environment**, check the variables were added, as indicated
  - Added – variable **TGTARM** has been set to **arm11**
  - Added – variable **BSP\_WATERMARK** has been set to **1**

2. In **Catalog Items** View, check that the following DirectShow system components of the OS Build were added or removed, as indicated:
  - Added - all components in **Core OS > CEBASE > Graphics and Multimedia Technologies -> Media -> DirectShow**
  - Added - the component **Core OS > CEBASE > Graphics and Multimedia Technologies > Media > Media Formats > MPEG-1 Parser/Splitter**
  - Added - the component **Core OS > CEBASE > Graphics and Multimedia Technologies > Media > Media Formats > AVI Filter**
  - Added - the component **Core OS > CEBASE > Graphics and Multimedia Technologies > Media > Audio Codecs and Renderers > Wave/AIFF/au/snd File Parser**

## 1.2.2 Removing an Existing Installed SDK

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

### NOTE

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

To remove an SDK installation, use these steps:

1. Close Platform Builder.
2. Click **Start > Settings > Control Panel > Add or remove Programs**.  
The Add or Remove Programs dialog is displayed.
3. Remove the SDK.
4. Manually remove the remaining SDK files and directories.

```
$(_WINCEROOT)\Platform\<platform_name>\src\Multimedia
```

## Chapter 2

# Building OS Images

After completing the SDK installation, you can use the sample workspace to build a Windows CE OS Image based on the installed SDK. You may add or remove image build components.

## 2.1 Working with Multimedia Components

The multimedia components are located in the Windows Catalog. You import the catalog, and can then add the components to the OS Image build.

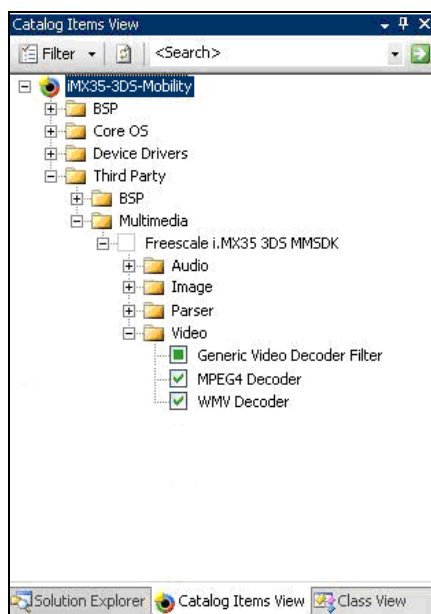
### 2.1.1 Importing the Windows Embedded CE 6.0 Catalog

For Windows Embedded CE 6.0, the multimedia catalog is automatically imported when the sample workspace is opened.

To view the catalog, use these steps:

1. Click **View > Other Windows > Catalog Items View**.

The Catalog Items View dialog is displayed (Figure 2-1).



**Figure 2-1 Windows Embedded CE 6.0 Catalog**

2. To view the multimedia components, open **Third Party > Multimedia > Freescale i.MX35 3DS MMSDK**.

If can not find these components in catalog items view, it need exit and restart the Platform Builder to make multimedia catalog imported completely.

## 2.1.2 Add/Remove components from Catalog

To add/remove components from the catalog, use these steps:

1. Click **View > Other Windows > Catalog Items View**.
2. To view the multimedia components, open **Third Party > Multimedia > Freescale i.MX35 3DS MMSDK**.
3. To add/remove a multimedia component, just select/unselect the associated check box.

## 2.2 Building OS Images

This section explains how to build Windows CE OS image included multimedia components in the sample workspace.

### 2.2.1 Building an Image in Platform Builder

For instructions for building OS images using Platform Builder, see the *Windows Embedded CE 6.0 for i.MX35 3-Stack User's Guide*.

### 2.2.2 Building an Image using the Command Line

The SDK installation provides the `fsl_mmfwk_build.bat` script, which builds the OS image. You can run this script during the last installation step or in the command line after installation.

To run the script in the command line, follow these steps:

1. Open the command console on your PC.
2. Go to the `$(_WINCEROOT)\platform\<platform>\src\Multimedia\` directory.
3. Run `fsl_mmfwk_build.bat`.

---

## 2.3 Building Multimedia Components in sample Workspace

This section explains how to build/rebuild installed multimedia components after the SDK upgrading or the modification on multimedia components' source code.

### 2.3.1 Building all components simultaneously

To build all of the installed components at once, use these steps:

1. Go to the Solution Explorer View, and then open the **Multimedia** project in the folder **Platform > iMX35-3DS-PDK1\_7 > SRC**.
2. Right-click on the **Multimedia** project, and then select **Rebuild**.

This builds all DLL and EXE binaries in the `bin600` directories under the Multimedia folder in the BSP code tree.

### 2.3.2 Building individual components

To build individual components, use these steps:

1. In the project window, open the individual component subproject in the **Multimedia** project
2. Right-click an individual component's sub-project, and then select **Rebuild**

This builds the DLL and EXE binaries in the `bin600` directories under the Multimedia folder in the BSP code tree.



## Chapter 3

# Using the Test Procedures

This chapter explains how to test the Freescale multimedia components in the Windows CE OS. The image was built as described in chapter 2.

### 3.1 Audio Decoder Test

To perform this test, use the Windows CE Media Player to playback the audio files.

### 3.2 Video Decoder Test

To perform this test, use the Windows CE Media Player to playback the video files.

### 3.3 Image Decoder Test

To test image decoders, use the test applications that are built into the Windows CE OS image when the decoder component is added from Catalog during the image build procedure. Double-click the image file to display the image.

Supported image file extensions:

- \*.bmp for BMP decoder
- \*.gif for GIF decoder
- \*.jpg for JPEG decoder
- \*.png for PNG decoder

Image display is also supported by command line. To display an image, run the associated command:

Image Type	Command
BMP	<code>fsl_bmp_img_lib_test.exe &lt;image_file_name&gt;</code>
GIF	<code>fsl_gif_img_lib_test.exe &lt;image_file_name&gt;</code>
JPEG	<code>fsl_jpeg_img_lib_test.exe &lt;image_file_name&gt;</code>
PNG	<code>fsl_jpeg_img_lib_test.exe &lt;image_file_name&gt;</code>

#### NOTE

---

If JPEG thumbnail image exists, JPEG image decoder test application will display thumbnail image after closing primary image display window.

## 3.4 BSAC Decoder Test

The test application for a BSAC audio decoder is built into the Windows CE OS image, when the BSAC decoder component is added from the Catalog during the image build procedure. The test application only supports the BSAC raw bit stream file input and 16bit stereo PCM output.

To execute the test application, run the following at the command line:

**fsl\_bsac\_dec\_filter\_test.exe** <input file> <output file>

Where the input arguments list below:

Argument	Description
<input file>	Input BSAC raw bit stream file name with full path.
<output file>	Output 16bit stereo PCM file name with full path

Refer to the following command line for information.

**fsl\_bsac\_dec\_filter\_test.exe rawdata.bsac output .pcm**

## 3.5 MP3 Encoder Test

The test application for an MP3 audio encoder is built into the Windows CE OS image, when the MP3 encoder component is added from the Catalog during the image build procedure. The test application supports the WAV file input.

To execute the test bench, run the following at the command line:

**fsl\_mp3\_enc\_dmo\_test.exe** <input file> <output file> <Encoder Configuration>

The <Encoder Configuration> parameters are optional, and described as follows:

**[*-b* <bps>] [*-s* <samplerate>]**

Arguments	Description
<b>-b &lt;bps&gt;</b>	Bit rate for the encoded mp3 bit stream (default 128000), bit/second in Unit.  Accepted values: 32000, 40000, 48000, 56000, 64000, 80000, 96000, 112000,

	128000, 160000, 192000, 224000, 256000, 320000
<b>-s &lt;samplerate&gt;</b>	Sample rate for the encoded mp3 bit stream (default 44100), sample/second in Unit  Accepted values: 32000, 44100, 48000

Refer to the following command line for information.

**fsl\_mp3\_enc\_dmo\_test.exe pcm.wav output.mp3 -b 128000 -s 44100**

### NOTE

The current test application supports the WAV file input. The sample rate of an input WAV should be the same as "**-s <samplerate>**". If the two rates are not identical, the test application stop the encode procedure and exits.

## 3.6 SBC Encoder Test

The test application for an SBC speech encoder is built into the Windows CE OS image when, the SBC encoder component is added from the Catalog during the image build procedure.

The SBC Encoder test application provides a set of arguments to set the configuration of the encoder.

To execute the test, run the following at the command line:

**fsl\_sbc\_enc\_test.exe < input\_file> [-h] [-l<blk\_len>] [-m<mode>] [-o<output\_file>] [-n<subbands>] [-p] [-r <rate>] [-b<bitpool>] [-s<sample\_freq>] [-f<super\_frame\_size>]**

Where the input arguments list below:

Argument	Description
<b>input_file</b>	Name of the input file
<b>-h</b>	Display this command line help and exit
<b>-l &lt;blk_len</b>	block length (4,8,12 or 16)
<b>-m &lt;mode&gt;</b>	mode (0=mono, 1=dual_channel, 2=stereo)
<b>-o output_file</b>	Name of the output file. Output suppressed if unspecified.
<b>-n &lt;subbands&gt;</b>	number of subbands (4 or 8)
<b>-p</b>	Enable psycho-acoustic model [default is off]
<b>-r &lt;rate</b>	Bit rate in bps (cannot be combined with -b option)
<b>-b &lt;bitpool&gt;</b>	Bit pool value (2 to 250) (cannot be combined with -r option)

	Recommended bit pool value: In DUAL MODE, if sample rate==16KHz, bitpool=5*subbands, if sample rate>16KHz,bitpool=4*subbands; In STEREO MODE, if sample rate==16KHz, bitpool=9*subbands; if Sample rate>16kHz,bitpool=7*subbands.
<b>-s</b> <b>&lt;sampling_freq&gt;</b>	Sampling Frequency (16000,32000,44100, or 48000 Hz)
<b>-f</b> <b>&lt;super_frame_size&gt;</b>	Size of input data frame (e.g. 2048) – optional

### NOTES

The current test application supports the WAV file input.

Either bit pool or bit rate must be provided, but not both.

Super frame size is optional. Encoder operates on data frame size that is a product of the number of subbands and number of blocks. If input data comes in frames of a certain predefined size, then the **super\_frame\_size quantity** should be set to that size.

Psychoacoustic model may or may not be used.

## 3.7 WMA8 Encoder Test

The test application for a WMA8 audio encoder is built into the Windows CE OS image, when the encoder component is added from the Catalog during the image build procedure.

To execute the test, run the following at the command line:

**fsl\_wma8\_enc\_dmo\_test.exe** *<input file>* *<output file>* *<Encoder Configuration>*

The parameters for Encoder Configuration are optional, and described as follows:

***[-a <AUTHOR>] [-b <BITRATE>] [-c <COPYRIGHT>] [-r <RATING>]***  
***[-d <DESCRIPTION>] [-T <TITLE>]***

The following table describes the parameters.

Argument	Description
<b>-a &lt;AUTHOR&gt;</b>	Specifies an array of WCHARs that contains the author

	information in the ASF header.
<i>-b &lt;BITRATE&gt;</i>	Bit rate for the encoded wma bit stream (default 70000). Any value ranging from 20k(bps) to 211.2k(bps).
<i>-c &lt;COPYRIGHT&gt;</i>	Specifies an array of WCHARs that contains the copyright information in the ASF header.
<i>-d &lt;DESCRIPTION&gt;</i>	Specifies an array of WCHARs that contains the description information in the ASF header.
<i>-r &lt;RATING&gt;</i>	Specifies an array of WCHARs that contains the rating information in the SF header.
<i>-T &lt;TITLE&gt;</i>	Specifies an array of WCHARs that contains the title information in the ASF header.
<i>-h</i>	Get the usage of this test bench.

The following table describes the valid configuration combination for WMA8 Encoder.

Refer to the following command line for information.

***fsl\_wma8\_enc\_dmo\_test.exe pcm.wav output.wma -b 128000 -d "Freescale test music"***

## 3.8 PEQ Audio Post-processing Test

The test applications for PEQ audio post-processing ***fsl\_peq\_ppp\_app\_forwmp.exe***, will be built into the WinCE OS image, when the post-processing component is added from Catalog during the image build procedure.

It is a graphic user interface for users to switch among PEQ profiles and control volume dynamically while playback the audio file with Windows CE Media Player.

### IMPORTANCE

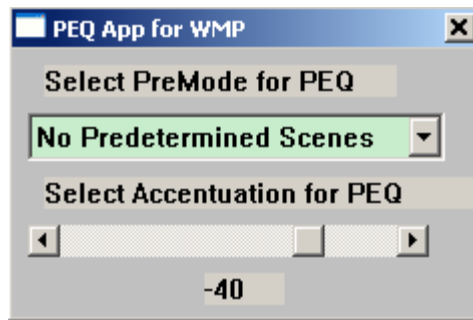
The test applications for PEQ audio post-processing should be run before using Windows CE Media Player to playback the audio files. Otherwise, PEQ audio post-processing will not function.

To use test application for PEQ testing, follow these steps:

1. To execute the test application, double-click the executable file or run the following at command line:

***fsl\_peq\_ppp\_app\_forwmp.exe***

Then the test application window is displayed (Figure 3-1)



**Figure 3-1 PEQ Setting**

2. Use Windows CE Media Player to playback an audio file. And keep the test application running when audio playback
3. During the playback periods, you can select different “**Premode**” dynamically through Profile List. The following table describes the Pre-defined profiles are defined.

Index	Profile Name
0	Do not select predetermined scenes. (default)
1	Acoustic
2	Bass booster
3	Bass reducer
4	Classical
5	Dance
6	Deep
7	Electronic
8	hip hop
9	Jazz
10	Latin
11	Loudness
12	Lounge
13	Piano
14	Pop
15	R&B
16	Rock
17	small speakers
18	spoken word
19	treble booster
20	treble reducer
21	vocal booster

22	Flat
----	------

4. Use **Attenuation** scrollbar to control the volume.

### NOTE

If the value for attenuation is set too high, the audio output would not be pleasing. It is strongly recommended that the value of attenuation is set to below -30.

## 3.9 G.726 Encoder/Decoder Test

G.726 encoder test application is built into the Windows CE OS image when the G.726 codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

**fsl\_g726\_enc\_test.exe** [-Options] <InFile> <OutFile>

The following table describes the arguments:

Argument	Description
<b>-Options</b>	<div>-law &lt;a u l&gt;      The letters A or a for G.711 A-law, letter u for G.711 m-law, or letter l for linear. Default is A-law.</div> <div>-rate #            the bit-rate (in kbit/s): 40, 32, 24 or 16. Default is 32kbit/s.</div> <div>-homing &lt;InitFile&gt;    The file contains initialization (homing) sequence to drive the Encoder to a known initial state. Default is no init file and the Codec is in reset state.</div> <div>-?/-help            print help message</div>
<b>InFile</b>	Name of the input file
<b>OutFile</b>	Name of the output file

G.726 decoder test application is built into the Windows CE OS image when the G.726 codec component is added from Catalog during the image build procedure.

To execute the test bench, run the following at the command line:

**fsl\_g726\_dec\_test.exe** [-Options] <InFile> <OutFile>

The following table describes the arguments:

Argument	Description
<b>-Options</b>	<div>-law &lt;a u l&gt;      The letters A or a for G.711 A-law, letter u for G.711 m-law, or letter l for linear. Default is A-law.</div>

	-rate #	the bit-rate (in kbit/s): 40, 32, 24 or 16. Default is 32kbit/s.
	-homing <InitFile>	The file contains initialization (homing) sequence to drive the Encoder to a known initial state. Default is no init file and the Codec is in reset state.
	-?/-help	print help message
<b>InFile</b>	Name of the input file	
<b>OutFile</b>	Name of the output file	

## 3.10 JPEG Encoder Test

The test application for the JPEG image encoder is built into the Windows CE OS image, when the encoder component is added from Catalog during the image build procedure.

The JPEG Encoder test application can output the encoded data to a file or render it to a screen. The input format of the test bench should be in YUV422 in interleaved format. The test application provides a set of the arguments to configure the encoder.

Run the following command in “Command Prompt” to execute the test bench:

```
fsl_jpeg_enc_img_lib_test.exe -yt <yuv_format> -q <quality> -yw <y_width> -yh
<y_height> -uw <u_width> -uh <u_hgeiht> -vw <v_width> -vh <v_hgeiht> [ -prg] [ -
rm] [ -ex] -i <input file > -o <output file>
```

The following table describes the parameters.

Argument	Description
<b>-yt &lt;yuv_format&gt;</b>	Accepted YUV formats: <ul style="list-style-type: none"> <li>3 – YU_YV_422_INTERLEAVED</li> <li>4 – YV_YU_422_INTERLEAVED</li> <li>5 – UY_VY_422_INTERLEAVED (default)</li> <li>6 – VY_UY_422_INTERLEAVED</li> </ul>
<b>-q &lt;quality&gt;</b>	Compress quality can be set the value in the range [0, 100]- (75 is default)
<b>-yw &lt;y_width&gt;</b>	Y width value in the range [1, 65000]
<b>-yh &lt;y_height&gt;</b>	Y height value in the range [1, 65000]
<b>-uw &lt;u_width&gt;</b>	U width value in the range [1, 65000]
<b>-uh &lt;u_height&gt;</b>	U height value in the range [1, 65000]
<b>-vw &lt;v_width&gt;</b>	V width value in the range [1, 65000]
<b>-vh &lt;v_height&gt;</b>	V height value in the range [1, 65000]
<b>-prg</b>	Enable Progressive Compress method



<b>-rm</b>	Enable Restart marker
<b>-ex</b>	Enable EXIF exist
<b>-i &lt;input file&gt;</b>	Input file name list and path
<b>-o &lt;output file&gt;</b>	Output file name and path

### NOTES

The current version of the library supports only the Interleaved YUV422 format; where the YUV format values are 3, 4, 5 and 6.

## 3.11 S/PDIF Transfer Test

The test application for the S/PDIF Transmitter is built into the Windows CE OS image, when the S/PDIF Transmitter component is added from Catalog during the image build procedure.

### IMPORTANCE

For S/PDIF Transfer Test, **Playback Device** must be switched to **SPDIF Output** via **audiorouting** program first.

Meanwhile S/PDIF Transmitter test application only supports AC3 compressed audio bit stream transmission.

For PCM raw data transmission, follow these steps:

1. Execute **audiorouting.exe** program which locates in **\Windows\** and switch **Playback Device** to **SPDIF Output**, just as Figure 3-2 shown.



**Figure 3-2 AudioRouting Setting**

2. Use the Windows CE Media Player to playback the audio files, then the decoded PCM raw data will be transmitted automatically through S/PDIF device. The transmitted data can be captured by S/PDIF RX device.

For AC3 compressed audio bit stream transmission, use S/PDIF Transmitter test application.

To use test application for S/PDIF Transmitter testing, follow these steps:

1. Execute **audiorouting.exe** program which locates in \Windows\ and switch **Playback Device** to **SPDIF Output**.
2. Execute S/PDIF Transmitter test application, run the following at the command line:

**fsl\_spdif\_tx\_testbench.exe** <TX file>

The following table describes the arguments:

Argument	Description
<TX file>	Name of the file whose AC3 compressed audio bit stream needs to be transmitted

Refer to the following command line for information.

**fsl\_spdif\_tx\_testbench.exe test.ac3**

## NOTES

S/PDIF Transmitter test application packetizes AC3 compressed audio bit stream according to IEC1937 frame formats by S/PDIF TX filter and transmits IEC1937 frames through S/PDIF device.

## 3.12 S/PDIF Receive Test

The test application for the S/PDIF Receiver is built into the Windows CE OS image, when the S/PDIF Receiver component is added from Catalog during the image build procedure.

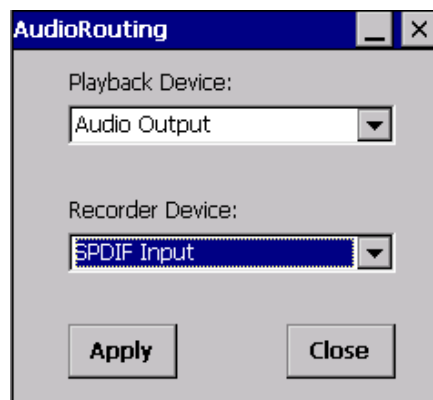
S/PDIF Receiver test application captures the PCM raw data or non-PCM compressed audio bit stream which is captured by S/PDIF device and saves them into WAV file or DAT file according to captured data is PCM raw data or not. In test application, S/PDIF RX filter distinguishes between PCM raw data or non-PCM compressed audio bit stream and de-packetizes non-PCM compressed audio bit stream from IEC1937 frame formats.

## IMPORTANCE

Before running S/PDIF Receiver test application, **Recorder Input** must be switched to **SPDIF Input** via **audiorouting** program.

To use test application for S/PDIF Receiver testing, follow these steps:

1. Execute **audiorouting.exe** program which locates in **\Windows\** and switch **Recorder Device** to **SPDIF Input**, just as Figure 3-3 shown.



**Figure 3-3 AudioRouting Setting**

2. Execute S/PDIF Receiver test application, run the following at the command line:

**fsl\_spdif\_dar\_app.exe** *<Time in sec>* *<Output file>*

The following table describes the arguments:

Argument	Description
<b>&lt;Time in sec&gt;</b>	Duration for audio capture, second in Unit.
<b>&lt;Output file&gt;</b>	Name of the output file

Refer to the following command line for information.

**fsl\_spdif\_dar\_app.exe 5 output**

### NOTES

S/PDIF Receiver test application automatically adds output file's extension name -- ".wav" for PCM raw data and ".dat" for non-PCM compressed audio bit streams.

Meanwhile S/PDIF Receiver test application doesn't support date type dynamic change at the process of transmission.

## 3.13 Deinterlacer Enable/Disable

The Video Post-processor Deinterlacer can be enabled or disabled using the registry key

```
[HKEY_LOCAL_MACHINE\Software\FreeScale\Filter\Video\PostProcess\deinterlace]  
Deinterlace=00000000
```

---

where:

"deinterlace"=dword:00000001 means deinterlacer enable; otherwise it is disabled.

## 3.14 DownMix Audio Post-processing Test

The test application for the DMX audio post-processing is built into the Windows CE OS image, when the DMX post-processing component is added from Catalog during the image build procedure.

The DMX post-processing implements the Multi-Channel audio down-mixing from up to 7.1 channels to Stereo. The input audio format should be in Multi-Channel PCM in interleaved format.

To execute the test, run the following at the command line,

**downmix\_dmo\_test.exe** <InputFile>

The <InputFile> would be Multi-Channel wav (PCM) file, and output audio would be sent to audio render. You can also play a multi-channel wave file with CE Player.

## 3.15 ASRC Post-processing Test

The test application for ASRC audio post-processing is built into the Windows CE OS image, when the *ASRC post-processing* component is added from the Catalog during the image build procedure. ASRC filter implements audio sample rate convention from one frequency to another.

### Notes

Input sample rate range is [8000Hz, 96000Hz].

Output sample rate range is [32000Hz, 96000Hz]

Due to the ASRC hardware limitation, maximum input audio channel is up to 2.

ASRC test application could implement ASRC in both file playback mode and stream playback mode.

For file playback mode, test application would input a WAV file, use ASRC to do the sample rate convention and then store data to a WAV file or output to audio render.

For stream playback mode, test application would record audio from S/PDIF port, use ASRC to do the sample rate convention and then store data to a WAV file or output to audio port.

To execute these tests, run the following at the command line:

**fsl\_asrc\_filter\_testbench.exe -m <mode> -i <input file> -s <samplerate> -t <time> -o <output file>**

The following table describes the arguments:

Argument	Description
<b>-m &lt;mode&gt;</b>	Playback Mode 1 – File playback (as default) 2 – Stream playback
<b>-i &lt;input file&gt;</b>	Name of the input WAV file – sample rate should be in the range [8000, 96000] – Valid for File Playback Mode only
<b>-s &lt;samplerate&gt;</b>	Output Sample rate (44100 as default) Accepted values: 32000, 44100, 48000, 64000, 88200, 96000
<b>-t &lt;time&gt;</b>	Playback time – unit in second – Valid for Stream Playback Mode only
<b>-o &lt;output file&gt;</b>	Name of the output wave file

Refer to the following command lines for information.

1. File playback testing (output to another WAV)

**fsl\_asrc\_filter\_testbench.exe -m 1 -i input\_48KHz.wav -s 44100 -o output\_44KHz.wav**

2. File playback testing (output to audio port)

**fsl\_asrc\_filter\_testbench.exe -m 1 -i input\_48KHz.wav**

3. Streaming playback testing (record from S/PDIF port and output to another WAV)

**fsl\_asrc\_filter\_testbench.exe -m 2 -t 10 -s 44100 -o output\_44KHz.wav**

4. Streaming playback testing (record from S/PDIF port and output to audio port)

**fsl\_asrc\_filter\_testbench.exe -m 2 -t 10**

### IMPORTANCE

Before start steaming playback testing, **Recorder Devices** must be switched to **SPDIF Input** via **AudioRouting** program locates in **\Windows\**



## Appendix A

### Patches and Fixes

Upon installation, Freescale patches are automatically installed. However, you may also install the patches manually, if you prefer to confirm that they are installed.

#### A.1 Unaligned Access Fault Disable

**Applies to:** Windows Embedded CE 6.0

**About:** The Freescale video decoder requires that the BSP provide memory unaligned access without fault.

**Action:** Before you build the BSP, perform the following modification:

1. Open the `bsp_cfg.h` file, located in: `$(_WINCEROOT)\PLATFORM\<platform>\src\inc\`
2. Uncomment the following line, so that it appears as follows:

```
#define BSP_OAL_DISABLE_ALIGNMENT_FAULT
```

#### A.2 Video Memory Cache Policy

**Applies to:** Windows Embedded CE 6.0

**About:** The Freescale video decoder requires that the cache policy of the video memory (IPU buffer) be write-through. Performance issues will occur with non-cacheable or write-back memory.

**Action:** Before you build the BSP, perform the following modification:

1. Open the `bsp_cfg.h` file in the following directory  
`\WINCE600\PLATFORM\<platform>\src\inc\`
2. In the file, change `BSP_VID_MEM_CACHE_WRITETHROUGH` to `TRUE`, as shown:

```
#define BSP_VID_MEM_CACHE_WRITETHROUGH TRUE
```

## Appendix B

# WinCE Kernel Debugger Configuration

When use WinCE debugger, some multimedia components might generate exceptions on their system check initialization but are safe to continue since the exceptions are handled directly by the multimedia components. This might disturb your debug environment with processing these exceptions.

The following steps specify how to configure your debugger so that these exceptions are handled automatically without user input needed:

1. From the Debug menu, click **Exception...**

The **Exceptions dialog box** opens.

2. In the **Exceptions** list, use the list control to select the individual exception or category of exceptions whose handling you want to change. The **Exceptions dialog box** then displays the exception number, description and action for that exception. Select **Add new**:

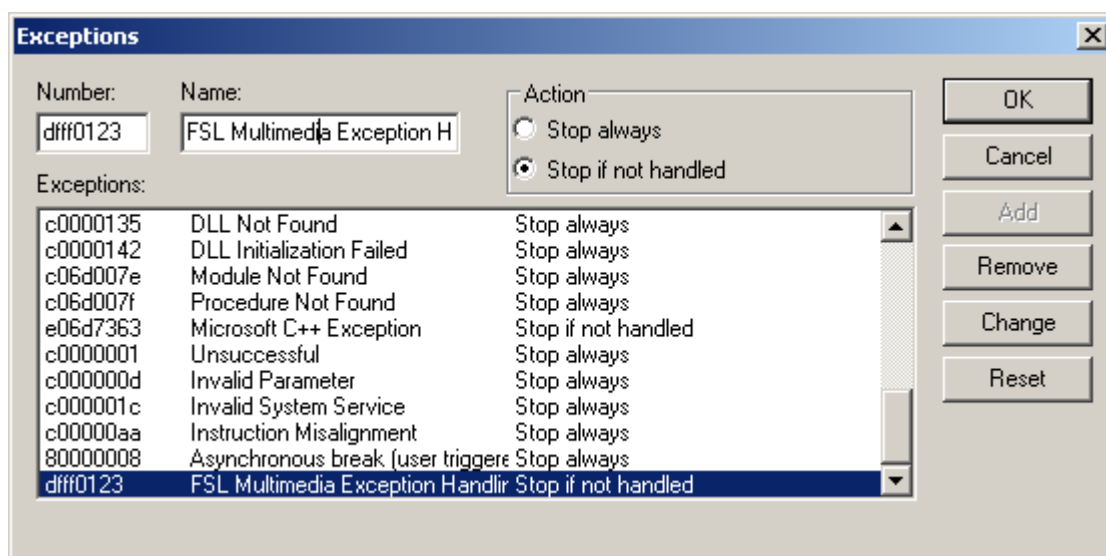


Figure 3-4 Watermark exception configuration window

- In field **Number**, input *DFFF0123*;
- In field **Name**, input *FSL Multimedia Exception Handling*;
- In field **Action Select**, select *Stop if Not Handled*;
- Click **Add**





- Click **OK**.