



06-3732-RN-TX30  
APRIL 3, 2006

5.1

# Release Notes for JPEG Encoder on ARM11 ELINUX

**ABSTRACT:**

Release Notes for JPEG Encoder on ARM11 ELINUX

**KEYWORDS:**

Multimedia codecs, JPEG, Image

**APPROVED:**

Wang Zening

## Revision History

Version	Date	Author	Change Description
0.1	22-Mar-2004	Harsha Deeph G	Engineering Release 0.1 of JPEG Encoder on eLinux platform
0.2	03-May-2004	Harsha Deeph G	Engineering Release 0.2 of JPEG Encoder on eLinux platform
0.3	13-May-2004	Harsha Deeph G	Engineering Release 0.3 of JPEG Encoder on eLinux platform
0.4	14-Jun-2004	Harsha Deeph G	Optimized Release 1.0
2.0	12-Oct-2004	Harsha Deeph G	ARM11 Release 2.0
3.0	08-Jan-2004	Harsha Deeph G	ARM11 Release 3.0
4.0	28-Feb-2005	Harsha Deeph G	ARM11 Release 4.0.(Tested on board)
4.1	08-Sep-2005	Anirudh Radhakrishnan	Build procedure changes for RVDS2.2
5.0	06-Feb-2006	Lauren Post	Using new format
5.1	03-April-2006	Sriram Shankar	Document review

# Table of Contents

<b>Introduction .....</b>	<b>4</b>
1.1 Purpose .....	4
1.2 Scope .....	4
1.3 Audience Description .....	4
1.4 References .....	4
1.4.1 Standards .....	4
1.4.2 General References .....	4
1.4.3 Freescale Multimedia References .....	4
1.5 Definitions, Acronyms, and Abbreviations .....	5
1.6 Document Location .....	5
<b>2 Release History .....</b>	<b>6</b>
2.1 Assumptions and Known Problems .....	7
2.2 Contacts .....	7
<b>3 List of Deliverables .....</b>	<b>8</b>
3.1 Documentation .....	8
3.2 Public Headers .....	8
3.3 Test Application Source .....	8
3.4 Library Source .....	8
3.5 Common Makefiles .....	9
3.6 Test Vectors .....	10
<b>4 Software Setup &amp; Tools used .....</b>	<b>11</b>
<b>5 Build Procedure .....</b>	<b>12</b>
5.1 Library .....	12
5.2 Test Application .....	13
<b>6 Test Application Execution .....</b>	<b>15</b>
6.1 Scripts .....	<b>Error! Bookmark not defined.</b>
6.2 ELINUX .....	<b>Error! Bookmark not defined.</b>
6.3 RVDS .....	<b>Error! Bookmark not defined.</b>
6.4 UNIX Reference .....	<b>Error! Bookmark not defined.</b>
<b>7 Pre compilation Options .....</b>	<b>16</b>

# Introduction

## 1.1 Purpose

The purpose of this document is to provide information on the package contents, instructions on building library and test applications and test execution on ARM11 ELINUX, RVDS and Linux x86.

## 1.2 Scope

The scope is restricted to information on the package contents and instructions for building and testing. This document does not provide architecture or details about the APIs provided in the package. Performance data will be provided in another document as detailed in the Requirements Book.

## 1.3 Audience Description

The reader is expected to have basic understanding of Image processing and JPEG Baseline encoding.

## 1.4 References

### 1.4.1 Standards

- DIS 10918-1 and draft DIS 10918-2
- "JPEG Still Image Data Compression Standard" by William B. Penne baker and Joan L. Mitchell published by Van No strand Reinhold, 1993, ISBN 0-442-01272-1. 638 pages, price US\$59.95. This book includes the complete text of the ISO JPEG standards (DIS 10918-1 and draft DIS 10918-2).

### 1.4.2 General References

- Wallace, Gregory K. "The JPEG Still Picture Compression Standard", Communications of the ACM, April 1991 (vol. 34 no. 4), pp. 30-44.

### 1.4.3 Freescale Multimedia References

- JPEG Encoder Application Programming Interface – jpeg\_enc\_api.doc
- JPEG Encoder Requirements Book - jpeg\_enc\_reqb.doc
- JPEG Encoder Test Plan - jpeg\_enc\_test\_plan.doc
- JPEG Encoder Release notes - jpeg\_enc\_release\_notes.doc
- JPEG Encoder Test Results – jpeg\_enc\_test\_results.doc
- JPEG Encoder Performance Results – jpeg\_enc\_perf\_results.doc
- JPEG Encoder Interface header – jpeg\_enc\_interface.h

- JPEG Encoder Test Application – jpeg\_enc\_app.c

## 1.5 Definitions, Acronyms, and Abbreviations

TERM/ACRONYM	DEFINITION
API	Application Programming Interface
ARM	Advanced RISC Machine
Data Unit	JPEG proposal defines a data unit as a sample in predictive codecs and a [8x8] block in case of DCT based codecs
DCT	Discrete Cosine Transform
FSL	Freescale
IDCT	Inverse Discrete Cosine Transform - Transform used to convert samples from Frequency Domain to Spatial Domain
IJG	Independent JPEG Group
JPEG	Joint Photographic Experts Group
MCU	Minimum Coded unit. JPEG proposal defines an MCU as the smallest group of interleaved data units
RVDS	ARM RealView Development Suite
TBD	To Be Determined
UNIX	Linux PC x/86 C-reference binaries

## 1.6 Document Location

docs/jpeg\_enc

## 2 Release History

RELEASE NUMBER	DELIVERABLES	FEATURES
3.0	<ul style="list-style-type: none"> <li>• Documentation</li> <li>• Application Interface header file</li> <li>• ELINUX and RVDS libraries and test applications</li> <li>• UNIX/Linux x/86 Reference library and test application</li> <li>• Makefiles and Source code for library and test application including optimized assembler for the ELINUX and RVDS libraries.</li> <li>• Test vectors</li> </ul>	<ul style="list-style-type: none"> <li>• Supports non interleaved YUV in separate buffers. Can be configured for either 420 or 422 or 444 YUV Input</li> <li>• Supports YUV 422 interleaved input. Four different formats will be supported</li> <li>• The encoder can be configured for various quality factors. [0,100]. The higher this number the better will be the encoded quality and more will be the bpp taken by the encoder. (bpp - bits per pixel)</li> <li>• The encoder can be configured to put restart markers into the JPEG bit-stream.</li> <li>• The encoder has the capability to encode image heights and widths that are not multiple of 8. The encoder supports any image height and width</li> <li>• The encoder returns error codes for the errors made by application in setting up parameters and in passing memory pointers.</li> <li>• EXIF file format supported</li> <li>• Compressed Thumbnails supported. Both JFIF and EXIF thumbnails are supported</li> <li>• Streaming Output supported</li> <li>• Frame Level and Row Level APIs supported</li> <li>• Progressive encoding supported</li> <li>• Debug Logs supported</li> <li>• Suspension supported</li> <li>• Support for both Little Endian and Big Endian Memory systems</li> </ul>
4.0	<ul style="list-style-type: none"> <li>• Makefiles and project files to compile, link and test libraries on board</li> </ul>	<ul style="list-style-type: none"> <li>• This release contains all the features of the previous release 3.0 and is tested on target Evaluation Board (With eLinux running on it)</li> </ul>
4.1	Same	<ul style="list-style-type: none"> <li>• Mentioned the time/cycles taken on board in the test results doc</li> <li>• Put an #ifdef in jerror.h to let the codec library link without any logging function definitions</li> </ul>
4.2	Same	<ul style="list-style-type: none"> <li>• Shared library support</li> <li>• Upgrade to RVDS 2.2</li> </ul>

**Table 1. Details of the release**

## 2.1 Assumptions and Known Problems

None

## 2.2 Contacts

Please report any problems to the following email address: [mmsw@freescale.com](mailto:mmsw@freescale.com)

## 3 List of Deliverables

### 3.1 Documentation

**Base directory:** /ARM11/

Subdirectory	Files
docs/jpeg_enc	jpeg_enc_api.doc jpeg_enc_reqb.doc jpeg_enc_test_plan.doc jpeg_enc_test_results.doc jpeg_enc_perf_results.doc jpeg_enc_release_notes.doc

### 3.2 Public Headers

**Base directory:** /ARM11/

Subdirectory	File
API_include	jpeg_enc_interface.h

### 3.3 Test Application Source

**Base directory:** /ARM11/src/image/jpeg\_enc

Subdirectory	Files
test/	“Makefile” makefile for building RVDS, UNIX and ELINUX board executables.
test/c_source	*.c, a) application code. b) Debug log code. It contains implementation of functions used for logging data and messages.
test/test_util/scripts	Batch files to be run on the board

### 3.4 Library Source

**Base directory:** /ARM11/src/image/jpeg\_enc

Subdirectory	Files
library	Makefile “Makefile” for building RVDS, UNIX, and ELINUX libraries. libjpeg_enc_arm11_RVDS.a – Special options for simulator testing libjpeg_enc_arm11_ELINUX.a - static library for board libjpeg_enc_arm11_ELINUX.so – shared library for board libjpeg_enc_UNIX.a – library for Linux x/86 – c reference code

library/c_source	*.c, JPEG encoder source code
library/asm_source	*.s, JPEG encoder assembler code
library/include	*.h, JPEG encoder library header files

## 3.5 Common Makefiles

**Base Directory:** /ARM11/common

Makefile	Description
common.mk	<p>This is a common makefile included in the codec library makefile for building the libraries. This file includes common options used by all codecs. Following flags can be overwritten or added to in the codec library makefile</p> <ol style="list-style-type: none"> <li>1. Path to toolchain tools (TC_ROOT)</li> <li>2. GNU header file path (HEADER_PATHS)</li> <li>3. GNU library path (LIB_PATHS)</li> <li>4. GNU Compiler/Assembler Options (GNU_CFLAGS, GNU_AFLAGS)</li> <li>5. Endian Flags</li> <li>6. Optimization Flags(OPTIM_LEVEL, OPTIM_TYPE)</li> <li>7. Common options for RVDS,UNIX and ELINUX (CFLAGS,AFLAGS)</li> <li>8. Build specific flags</li> <li>9. Source directory of 'C' code</li> <li>10. Source directory of 'assembly(.s)' code</li> <li>11. Object directory for .o files</li> <li>12. RVDS Compilation Tools</li> <li>13. Codec header path</li> <li>14. Arguments for librarian for UNIX builds</li> <li>15. SHARED_ELINUX builds for libraries that must be linked using the toolchain because of external library includes.</li> </ol>
common_testapp.mk	<p>This is the common makefile included in the codec test makefile for building the test application. This file includes the common options used by the all the codecs. Following flags can be overwritten or added to in the codec test makefile</p> <ol style="list-style-type: none"> <li>1. Toolchain path depending on the build option</li> <li>2. Compiler Flags</li> <li>3. Linker flags</li> <li>4. Paths for c_source, exe and object directories</li> <li>5. Codec header files' INCLUDES path</li> <li>6. Endian Flags</li> <li>7. CODEC_LIB generation</li> </ol>

## 3.6 Test Vectors

**Base Directory:** multimedia\_vectors/test\_vectors

The test vectors are provide in another location

Directory	File	Description
jpeg_enc/input	420/* 422/* 444/* INTERLEAVED/422/*	Files that can contain YUV Image data. '420', '422' and '444' directories contain non interleaved input data. 'INTERLEAVED/422' directory contains interleaved 422 data
jpeg_enc/input/bmp	*.bmp	BMP files. YUV input files are generated from these bmp files. These are provided in this package just to enable the user/tester of this release to view the input. These files are not given as input to the encoder.
jpeg_enc/ref	*.jpg	Previously generated output files that can be taken as reference

## 4 Software Setup & Tools used

- ARM RVDS 2.2 (build 503) should be installed in the PC.
- Freescale Linux OS Release L26.1.15 must be running on the evaluation board.
- Intel based Red Hat Linux Machine must have the Montavista toolchain installed on it.
  - MontaVista 3.4.3-25.0.36.0501313 2005-08-21
- ‘Cygwin’ **Version CYGWIN\_NT-5.1**, a freely downloadable linux emulator is installed in PC - <http://www.cygwin.com/>.
- ‘make’ utility available for targeted platforms

## 5 Build Procedure

All the required makefiles are provided under individual directories. The library can be built for windows / target processor (ARM1136J-S). The details for the build procedure are described below.

### 5.1 Library

To build the library, run ‘make’ on ‘Makefile’ from library directory. The makefile shall create the required directory to hold the object files. The makefile can be used if you want to build the library only. The same makefile can be used to build libraries for both board, Unix/Linux and RVDS with different build options. The following options are available to build the library.

#### Options

##### a) BUILD options:

- **BUILD=ELINUX**: This is the default option and builds both static library ‘libjpeg\_enc\_arm11\_ELINUX.a’ and shared library ‘libjpeg\_enc\_arm11\_ELINUX.so’, for testing on the board.
- **BUILD=RVDS**: This option builds the static library ‘libjpeg\_enc\_arm11\_RVDS.a’, for testing on RVDS (Armulator).
- **BUILD=UNIX**: This option builds the static library ‘libjpeg\_enc\_UNIX.a’, for testing on UNIX/Linux machine.

**Eg:**  
make BUILD=ELINUX  
make BUILD=RVDS  
make BUILD=UNIX

##### b) ENDIAN options for RVDS:

- **TARGET\_ENDIAN=LITTLE**: This is the default option and sets the endian-ness to ‘little’
- **TARGET\_ENDIAN=BIG**: This option sets the endian-ness to big  
**Eg:** make BUILD=RVDS TARGET\_ENDIAN=BIG

##### c) clean options:

- **clean\_RVDS**: Deletes all the object files and the RVDS library ‘libjpeg\_enc\_arm11\_RVDS.a’.
- **clean\_ELINUX**: Deletes all the object file and the ELINUX libraries libjpeg\_enc\_arm11\_ELINUX.a and libjpeg\_enc\_arm11\_ELINUX.so.
- **clean\_UNIX**: Deletes all the object files and the UNIX library ‘libjpeg\_enc\_UNIX.a’.
- **clean**: Deletes all the object files and RVDS, UNIX and ELINUX libraries.

**Note:** Make appropriate changes in file ‘common.mk’ at directory ‘ARM11/common’ for the location of toolchains.

The library that is built is saved as `libjpeg_enc_arm11_RVDS.a` for RVDS build, and `libjpeg_enc_arm11_ELINUX.a` and `libjpeg_enc_arm11_ELINUX.so` for board build. These libraries are saved in the current directory (the same directory in which the source and assembly directories are listed).

Target	Compilation Environment	Build Options	Library Name
Board	PC (Using Cygwin)	BUILD= ELINUX	libjpeg_enc_arm11_ELINUX.a libjpeg_enc_arm11_ELINUX.so
RVDS	PC (Using Cygwin)	BUILD=RVDS TARGET_ENDIAN= BIG/LITTLE	libjpeg_enc_arm11_RVDS.a
Unix/ Linux	Unix/Linux machine	BUILD=RVDS TARGET_ENDIAN= BIG/LITTLE	libjpeg_enc_UNIX.a

## 5.2 Test Application

To build the test application, run 'make' on 'Makefile' from the test directory. This makefile can create executables for testing on Linux x86, the ARM11board and RVDS for ARM11. The executables `jpeg_enc_arm11_RVDS` for RVDS, `jpeg_enc_arm11_ELINUX` for board and `jpeg_enc_UNIX` for UNIX are stored under test/exe directory. The makefile shall create the required directory structure to hold the object files and executables. The following commands should be invoked so as to build the executables.

### Options

#### 1) BUILD options:

- **BUILD=ELINUX:** This is the default option and builds the executable '`jpeg_enc_arm11_ELINUX`', for the board.
- **BUILD=RVDS:** This option builds the executable '`jpeg_enc_arm11_RVDS`' for the RVDS (Armulator).
- **BUILD=UNIX:** This option builds the executable '`jpeg_enc_UNIX`' for the Unix/Linux machine.

**Eg:**

```
make BUILD=ELINUX (for board)
make BUILD=RVDS   (for Armulator)
make BUILD=UNIX   (for Unix/Linux machine)
```

#### 2) ENDIAN options for RVDS:

- **TARGET\_ENDIAN=LITTLE:** This is the default option and sets the endian-ness to 'little'

- **TARGET\_ENDIAN=BIG:** This option sets the endian-ness to big  
**Eg:** make BUILD=RVDS TARGET\_ENDIAN=BIG

### 3) **LIBRARY options:**

- **LIB=STATIC:** This option builds the ELINUX test application linked with the ELINUX static library 'libjpeg\_enc\_arm11\_ELINUX.a'. If nothing is specified, the executable links with shared library 'libjpeg\_enc\_arm11\_ELINUX.so'

**Eg:** make LIB=STATIC

### 4) **clean options:**

- **clean\_RVDS:** Deletes all the object files and the RVDS executable 'jpeg\_enc\_arm11\_RVDS'.
- **clean\_ELINUX:** Deletes all the object file and the ELINUX 'jpeg\_enc\_arm11\_ELINUX'.
- **clean\_UNIX:** Deletes all the object files and the Unix/Linux executable 'jpeg\_enc\_UNIX'.
- **clean:** Deletes all the object files and RVDS, UNIX ELINUX executables.

### **Note:**

In 'common\_testapp.mk' at directory 'ARM11/common', the paths for the compiling and linking tools are hard coded for the current set-up. These paths may not be the same in the user's directory set up. Hence, the 'common\_testapp.mk' should be modified to point to the directories where the linking and compilation tools are present before building the application for board.

The following table summarises the build options,

<b>Target</b>	<b>Compilation Environment</b>	<b>Build Options</b>	<b>Executable Name</b>
Board	Redhat Linux Machine	BUILD=ELINUX LIB=STATIC	jpeg_enc_arm11_ELINUX
RVDS	PC (Using Cygwin)	BUILD=RVDS TARGET_ENDIAN=LITTLE/BIG	jpeg_enc_arm11_RVDS
UNIX/Linux	Unix/Linux machine	BUILD=UNIX TARGET_ENDIAN=LITTLE/BIG	jpeg_enc_UNIX

## 6 Test Application Execution

To know the options provided by the test application, run the executable with ‘-h’ argument. It shall print a brief summary of all the options available.

### 6.1 ELINUX test APP

```
test_jpeg_enc_arm11_elinux -yt <yuv_format> -q <quality> -rm <reset marker> -yw  
<y_width> -yh <y_height> -uw <u_width> -uh <u_height> -vw <v_width> -vh <v_height>  
-ni <y input file> <u input file> <v input file> -o <output file> -ex <exif flag> [-prg]
```

for yuv\_format:

- 0: indicate YUV\_444\_NONINTERLEAVED,
- 1: indicate YUV\_422\_NONINTERLEAVED,
- 2: indicate YUV\_420\_NONINTERLEAVED,
- 3: indicate YU\_YV\_422\_INTERLEAVED,
- 4: indicate YV\_YU\_422\_INTERLEAVED,
- 5: indicate UY\_VY\_422\_INTERLEAVED,
- 6: indicate VY\_UY\_422\_INTERLEAVED

[-prg]: progressive compress method will be used if this flag is added into command. Otherwise, sequential method will be used

Example:

```
./test_jpeg_enc_arm11_elinux -prg -yt 1 -q 75 -rm 0 -yw 220 -yh 140 -uw 110 -uh 140 -vw  
110 -vh 140 -ni input/422/rose_220x140422.ycomp input/422/rose_220x140422.ucomp  
input/422/rose_220x140422.vcomp -o rose_220x140422_pro_linux_dut.jpg
```

## 7 Pre compilation Options

The following C options need to be set

<b>C Defines</b>	<b>Description</b>	<b>Remarks</b>
LOG_TIMING	To log performance timing results	