

# **Digital Video Test Bench**

**Version 4.20.18**

**Release Notes**

**22 March 2011**

The Digital Video Test Bench is a package containing a set of tools to facilitate verification and validation of the DVEVM/DVSDK product. DVTB provides support for executing functionality test cases involving end-to-end data flows. It also allows users to control and configure audio and video devices as well as all codec classes (via the Codec Engine) targeted for DVEVM.

---

**TABLE OF CONTENTS**

---

<b>1</b>	<b>New In This Release.....</b>	<b>3</b>
<b>2</b>	<b>System Requirements .....</b>	<b>3</b>
<b>3</b>	<b>Installing DVTB .....</b>	<b>3</b>
<b>4</b>	<b>Uninstall DVTB .....</b>	<b>9</b>
<b>5</b>	<b>Release Contents.....</b>	<b>9</b>
<b>6</b>	<b>Fixed In This Release .....</b>	<b>12</b>
<b>7</b>	<b>Known Issues in DVTB .....</b>	<b>12</b>
<b>8</b>	<b>DVTB Limitation .....</b>	<b>13</b>
<b>9</b>	<b>Known Issues in dependent components .....</b>	<b>15</b>
<b>10</b>	<b>Limitation in dependent components .....</b>	<b>16</b>

## **1 New In This Release**

- In DVTB Makefiles, libnurses is replaced with libncursesw.

## **2 System Requirements**

Please refer to DVEVM/DVSDK Technical Requirements Specification for the requirements for this package.

## **3 Installing DVTB**

### **DM6446(Linux)**

- Go to `$(DVSDK_INSTALL_DIR)/dvtb_X_XX_XX`
- Run `"make clean ; make dm6446"`
- Release (dvtb-r) and Debug (dvtb-d) binaries will be created under the following folder:  
`$(DVSDK_INSTALL_DIR)/dvtb_X_XX_XX/packages/ti/sdo/dvtb/dm6446/bin`
- Copy the binaries (dvtb-r & dvtb-d) to your target execution environment

### **DM355(Linux)**

- Go to `$(DVSDK_INSTALL_DIR)/dvtb_X_XX_XX`
- Run `"make clean ; make dm355"`
- Release (dvtb-r) and Debug (dvtb-d) binaries will be created under the following folder:  
`$(DVSDK_INSTALL_DIR)/dvtb_X_XX_XX/packages/ti/sdo/dvtb/dm355/bin`
- Copy the binaries (dvtb-r & dvtb-d) to your target execution environment

### **DM6467(Linux)**

- Go to `$(DVSDK_INSTALL_DIR)/dvtb_X_XX_XX`
- To build for TSPA codec server with h264 decoder (Ensure that the CODEC\_INSTALL\_DIR is pointing to TSPA codec server)
- Run `"make clean ; make dm6467 CODECS=TSPA H264CodecPackage=h264dec"`
- To build for TSPA codec server with h264 1080p decoder (Ensure that the CODEC\_INSTALL\_DIR is pointing to TSPA codec server)
- Run `"make clean ; make dm6467 CODECS=TSPA H264CodecPackage=h2641080p60vdec"`
- To build for NON TSPA codec server with h264 decoder (Ensure that the CODEC\_INSTALL\_DIR is pointing to NON TSPA codec server)
- Run `"make clean ; make dm6467 CODECS=NONTSPA H264CodecPackage=h264dec"`

- To build for NON TSPA codec server with h264 1080p decoder (Ensure that the CODEC\_INSTALL\_DIR is pointing to NON TSPA codec server)
- Run "make clean ; make dm6467 CODECS=NONTSPA H264CodecPackage=h2641080p60vdec"
- Release (dvtb-r) and Debug (dvtb-d) binaries will be created under the following folder:  
`$(DVSDK_INSTALL_DIR)/dvtb_X_XX_XX/packages/ti/sdo/dvtb/dm6467/bin`
- Copy the binaries (dvtb-r & dvtb-d) to your target execution environment

### **DM365(Linux)**

- Go to `$(DVSDK_INSTALL_DIR)/dvtb_X_XX_XX`
- To build for TSPA codec server (Ensure that the CODEC\_INSTALL\_DIR is pointing to TSPA codec server)
- Run "make clean ; make dm365 CODECS=TSPA"
- To build for TSPA codec server (Ensure that the CODEC\_INSTALL\_DIR is pointing to TSPA codec server)
- Run "make clean ; make dm365 CODECS=NONTSPA"
- Release (dvtb-r) and Debug (dvtb-d) binaries will be created under the following folder:  
`$(DVSDK_INSTALL_DIR)/dvtb_X_XX_XX/packages/ti/sdo/dvtb/dm365/bin`
- Copy the binaries (dvtb-r & dvtb-d) to your target execution environment

### **OMAP3530(Linux)**

- Go to `$(DVSDK_INSTALL_DIR)/dvtb_X_XX_XX`
- Run "make clean ; make omap3530"
- Release (dvtb-r) and Debug (dvtb-d) binaries will be created under the following folder:  
`$(DVSDK_INSTALL_DIR)/dvtb_X_XX_XX/packages/ti/sdo/dvtb/omap3530/linux/bin`
- Copy the binaries (dvtb-r & dvtb-d) to your target execution environment

### **DM3730(Linux)**

- Go to `$(DVSDK_INSTALL_DIR)/dvtb_X_XX_XX`
- Run "make clean ; make dm3730"
- Release (dvtb-r) and Debug (dvtb-d) binaries will be created under the following folder:  
`$(DVSDK_INSTALL_DIR)/dvtb_X_XX_XX/packages/ti/sdo/dvtb/dm3730/linux/bin`
- Copy the binaries (dvtb-r & dvtb-d) to your target execution environment

---

**OMAP3530(WinCE)**


---

**▪ Create DVTB subproject**

- Open up your OS design in Visual Studio
- Create a new directory wince\_project/wince600/dvtb at {DVSDK\_INSTALL\_DIR}/dvtb\_x\_xx\_xx/packages/ti/sdo/dvtb/omap3530/wince, where {DVSDK\_INSTALL\_DIR} is the DVSDK installation directory.
- Select Project-> Add New Subproject from the menu. Select "WCE Console Application" under "Available templates". Set "Subproject name" as "dvtb" and "Location" as {DVSDK\_INSTALL\_DIR}/dvtb\_x\_xx\_xx/packages/ti/sdo/dvtb/omap3530/wince\_project/wince600/dvtb
- In Next Window select "An empty subproject".
- Go to folder {DVSDK\_INSTALL\_DIR}/dvtb\_x\_xx\_xx/packages/ti/sdo/dvtb/omap3530/wince\_project/wince600/dvtb
- Delete postlink.bat
- Append prelink.bat with following

```
@echo off

rem
=====

rem @file prelink.bat
rem
rem @path ${DVTB_INSTALL_DIR}/wince_project/wince600/dvtb
rem
rem @desc Sets the DVTB environment and builds DVTB lib. This
rem file is used by WinCE platform builder to build the DVTB subproject.
rem
rem @ver
rem
=====

rem Copyright (C) 2010 Texas Instruments Incorporated - http://www.ti.com/
rem
=====

@REM Add pre-link commands below.
@set VAR_NAMES_SET=_WINCEROOT _winceroot _PROJECTROOT _projectroot _TGTCPU
_tgtcpu
@set VAR_NAMES_SET=%VAR_NAMES_SET% _TGTCPUFAMILY _tgtcpufamily
_TARGETPLATROOT _targetplatroot
@set VAR_NAMES_SET=%VAR_NAMES_SET% _FLATRELEASEDIR _flatreleasedir
WINCEDEBUG wincedebg
```

```

@set VAR_NAMES_SET=%VAR_NAMES_SET% PATH path BIOS_INSTALL_DIR
bios_install_dir

@call :Loop4 %VAR_NAMES_SET%

@goto start_build

:Loop4
@if [%1]==[] goto:EOF
@call set my_temp=%%%2%%
@set %2=
@set %1=%my_temp%
@shift
@shift
@goto Loop4

:start_build
@set VAR_NAMES_SET=
@set my_temp=

echo
=====
==

echo Starting DMAI DVTB builds
echo
=====
==

echo Current dir
cd %CD%\..\..\..\..\..\..\
gmake clean
gmake omap3530_w
gmake omap3530_w_install
echo
=====
==

echo DVTB lib done
echo
=====
==

cd %_WINCEROOT%

```

- Remove the contains of sources and add following

```

_COMMONPUBROOT=$(_PROJECTROOT)\cesysgen
__PROJROOT=$(_PROJECTROOT)

```

```
RELEASETYPE=LOCAL

_ISVINCPATH=$(_WINCEROOT)\public\common\sdk\inc;
_OEMINCPATH=$(_WINCEROOT)\public\common\oak\inc;$(_WINCEROOT)\public\common\s
dk\inc;
TARGETNAME=dvtb
FILE_VIEW_ROOT_FOLDER= \
    ReadMe.txt \
    prelink.bat \

FILE_VIEW_RESOURCE_FOLDER= \

FILE_VIEW_INCLUDES_FOLDER= \

SOURCES= \

EXEENTRY=mainWCRTStartup
TARGETLIBS= \
    $(_PROJECTROOT)\cesysgen\sdk\lib\$_CPUINDPATH\coredll.lib \

TARGETTYPE=NOTARGET
PRELINK_PASS_CMD=prelink.bat
FILE_VIEW_PARAMETER_FOLDER= \
    ProjSysgen.bat \
```

➤ Close your OS Design

## ▪ Build DVTB

There are two methods for building DVTB. Either as a WinCE platform builder subproject or from WinCE platform builder command line.

### 1) Using Platform builder subproject

- In Windows, add the installation directory of the XDCTOOLS to your PATH environment variable if not already done.
- Include following in {DVSDK\_INSTALL\_DIR}/dvtb\_x\_xx\_xx/Makefile
- -include Rules.make
- If it is already included in Makefile then this step is not required.
- Open up your OS design in Visual Studio
- Select Project->Add existing subproject from the menu. Open {DVSDK\_INSTALL\_DIR}/dvtb\_x\_xx\_xx/packages/ti/sdo/dvtb/omap3530/

wince/wince\_project/wince600/dvtb/dvtb.pbpxml, where  
{DVSDK\_INSTALL\_DIR} is the DVSDK installation directory.

- Select Project->EVM\_3530 Properties. Under Configuration Properties->Environment, set the following environment variables (if not set already):
- XDC\_INSTALL\_DIR: The installation directory of the XDCTOOLS from TI
- XDCPATH: The xdcpath used by the configuro command invocation in the makefile. This path needs to point to the xdc packages of TI software(Codec Engine, Codec Engine cetools, DSPLINK, the codec server, xdctools) and cesysgen include files. Eg. XDCPATH= C:/ti\_tools/xdctools\_3\_10\_05\_61/packages;D:/DVTB\_ON\_WINCE/dvSDK/DVSDK\_WINCE\_01\_00\_00\_04/codec\_engine\_2\_24/packages;D:/DVTB\_ON\_WINCE/dvSDK/DVSDK\_WINCE\_01\_00\_00\_04/codec\_engine\_2\_24/cetools/packages;D:/DVTB\_ON\_WINCE/dvSDK/DVSDK\_WINCE\_01\_00\_00\_04/dsplink\_sla\_1\_6\_1\_04;D:/DVTB\_ON\_WINCE/dvSDK/DVSDK\_WINCE\_01\_00\_00\_04/cs1omap3530\_1\_00\_00/packages;C:\WINCE600\OSDesigns\EVM\_3530\EVM\_3530\Wince600\TI\_EVM\_3530\_ARMV4I\cesysgen\sdk\inc;
- Select Build->Advanced Build Commands->Build Current BSP and Subprojects. This will rebuild DVTB. Note that any errors when building the DVTB will not appear in the Build window. They will however be logged in the build.log file under the {DVSDK\_INSTALL\_DIR}/dvtb\_x\_xx\_xx/packages/ti/sdo/dvtb/omap3530/wince/wince\_project/wince600/dvtb/ directory. This is because the build is done using a batch file in the background
- Binaries are generated at
- {DVSDK\_INSTALL\_DIR}/dvtb\_X\_XX\_XX/packages/ti/sdo/dvtb/omap3530/wince/bin and copied to \$(\_FLATRELEASEDIR) as well.
- Any future rebuild of DVTB can be done either using the GUI by right-clicking on the dvtb subproject in the Solution Explorer or selecting "Build".
- load OS image onto the EVM. dvtb.exe will be loaded with OS image. Alternatively DVTB executables (dvtb.exe & dvtb-d.exe) can be copied to target.
  - From WinCE command window dvtb app can be run.

## **2) Using Platform builder command Line**

- In Windows, add the installation directory of the XDCTOOLS to your PATH environment variable if not already done.
- Include following in {DVSDK\_INSTALL\_DIR}/dvtb\_x\_xx\_xx/Makefile
- -include Rules.make
- If it is already included in Makefile then this step is not required.
- Open up your OS design in Visual Studio
- Select Build->Rebuild EVM\_3530. If BSP is already build then this is not required.



- Select Project->Add existing subproject from the menu. Open {DVSDK\_INSTALL\_DIR}/dvtb\_x\_xx\_xx/packages/ti/sdo/dvtb/omap3530/wince/wince\_project/wince600/dvtb/dvtb.pbpxml, where {DVSDK\_INSTALL\_DIR} is the DVSDK installation directory.
- Select Project->EVM\_3530 Properties. Under Configuration Properties->Environment, set the following environment variables (if not set already):
- XDC\_INSTALL\_DIR: The installation directory of the XDCTOOLS from TI
- XDCPATH: The xdcpath used by the configuro command invocation in the makefile. This path needs to point to the xdc packages of TI software (Codec Engine, Codec Engine cetools, DSPLINK, the codec server, xdctools) and cesysgen include files. E.g. XDCPATH= C:/ti\_tools/xdctools\_3\_10\_05\_61/packages;D:/DVTB\_ON\_WINCE/dvSDK/DVSDK\_WINCE\_01\_00\_00\_04/codec\_engine\_2\_24/packages;D:/DVTB\_ON\_WINCE/dvSDK/DVSDK\_WINCE\_01\_00\_00\_04/codec\_engine\_2\_24/cetools/packages;D:/DVTB\_ON\_WINCE/dvSDK/DVSDK\_WINCE\_01\_00\_00\_04/dsplink\_sla\_1\_6\_1\_04;D:/DVTB\_ON\_WINCE/dvSDK/DVSDK\_WINCE\_01\_00\_00\_04/cs1omap3530\_1\_00\_00/packages;C:\WINCE600\OSDesigns\EVM\_3530\EVM\_3530\Wince600\TI\_EVM\_3530\_ARMV4I\cesysgen\sdk\inc;
- Select "Build->Open Release Directory in Build window" and open up a build window.
- To build DVTB, change directory into the dvtb installation directory {DVSDK\_INSTALL\_DIR}/dvtb\_x\_xx\_xx/. Then run the following commands in the build window
- gmake clean
- gmake omap3530\_w
- Binaries are generated at
- {DVSDK\_INSTALL\_DIR}/dvtb\_X\_XX\_XX/packages/ti/sdo/dvtb/omap3530/wince/bin.
- Run following command to copy dvtb binaries at \${\_FLATRELEASEDIR}
- gmake omap3530\_w\_install
- Copy the executables (dvtb.exe & dvtb-d.exe) to your target execution environment.
- From WinCE command window dvtb app can be run.

**Note:** DVTB expects codec server to be available at root of file system represented by a backslash (\).

## **4 Uninstall DVTB**

N/A

## **5 Release Contents**

The DVTB package for this product supports the following platforms:

- DM6446(Linux)
- DM355(Linux)
- DM6467(Linux)
- DM365(Linux)
- OMAP3530(Linux)
- DM3730(Linux)
- OMAP3530(WinCE)

Platform-wise use-cases supported:

**DM6446**(Linux)

- Video Playback of H.264 BP, MPEG4 SP, MPEG2 MP
- Video Capture of H.264 BP, MPEG4 SP
- Audio Playback of AAC-HE, MPEG1-L2-L3
- Speech Capture of G.711
- Speech Playback of G.711
- Configuration of codec for decode/encode
- Decode H.264 BP, MPEG4 SP, MPEG2 MP to YUV 422 format and store
- Decode H.264 BP, MPEG4 SP, MPEG2 MP to YUV 420 format and store
- Decode AAC, MPEG1-L2 to PCM format and store
- Decode G.711 to PCM format and store
- Encode raw video in YUV 422 format with H.264 BP, MPEG4 SP and store
- Encode raw video in YUV 420 format with H.264 BP, MPEG4 SP and store
- Encode raw audio in PCM format with G.711 and store
- Simultaneous Playback of Video and Audio streams
- Simultaneous Playback of Video and Speech streams
- Simultaneous Capture and Playback of H.264 BP video
- Simultaneous Capture of Video and Speech streams

**DM355**(Linux)

- Configuration of codec for decode/encode
- Decode MPEG4 to YUV and store
- Decode G.711 to PCM format and store
- Encode raw video with MPEG4 and store
- Encode raw audio in PCM format with G.711 and store
- Encode raw image with JPEG and store
- Simultaneous Decode of Video and Speech streams
- Simultaneous Encode of Video and Speech streams

**DM6467**(Linux)

- Video Decode of H.264, MPEG2, MPEG4, H2641080p60vdec
- Video Encode of H.264, MPEG4, H264fhdvenc
- Audio Decode of AAC-HE
- Speech Encode of G.711
- Speech Decode of G.711
- Image encode of JPEG
- Image Decode of JPEG
- Configuration of codec for decode/encode

- Decode H.264, MPEG2, MPEG4, H2641080p60vdec to YUV 420SP format and store
- Decode AAC-HE to PCM format and store
- Decode G.711 to PCM format and store
- Encode raw video in YUV 420P format with H.264, MPEG4, H264fhdvenc and store
- Encode raw audio in PCM format with G.711 and store
- Encode raw image in YUV 422 format with JPEG and store
- Encode raw image in YUV 420 format with JPEG and store

**DM365(Linux)**

- Configuration of codec for decode/encode
- Video Decode of H.264, MPEG2, MPEG4, MPEG4HDTVICP,VC1
- Video read and Encode of H.264,MPEG2, MPEG4, MPEG4HDTVICP
- Audio Decode of AAC
- Audio Encode of AAC
- Speech Encode of G.711
- Speech Decode of G.711
- Image Read and encode of JPEG
- Image Decode of JPEG
- Decode H264, MPEG2, MPEG4, MPEG4HDTVICP and VC1 streams to YUV and store
- Decode AAC streams to PCM format and store
- Decoder G.711 to PCM format and store
- Decode JPEG image to YUV and store
- Encode raw video with MPEG4, and store
- Encode raw video in YUV 420SP format with MPEG4 H.264,MPEG2, MPEG4HDTVICP and store
- Encode PCM with AAC and G7.11 and store
- Encode raw image with JPEG and store

**OMAP3530(Linux)**

- Decode MPEG2, MPEG4 and H264 to YUV 420P/422ILE format and store
- Decode MPEG2, MPEG4 and H264 to YUV 422ILE format and display
- Decode AAC-LC and AAC-HE to PCM format and store/play
- Decode G.711 to PCM and store/play
- Decode JPEG to 422ILE format and display
- Decode JPEG and store
- Encode and dump raw video in 422I and 420P YUV format with MPEG4 SP and H264
- Encode and dump raw YUV with JPEG
- Read/capture PCM and encode, dump with G.711
- Simultaneous Decode and store/Play/display of Video and Audio/Speech streams
- Simultaneous Decode and store/display/play of Image and Audio/Speech streams
- Encode yuv to h264 and decode it back and dump the decoded output in a yuv file

---

**DM3730(Linux)**

- Decode MPEG2, MPEG4 and H264 to YUV 420P/422ILE format and store
- Decode MPEG2, MPEG4 and H264 to YUV 422ILE format and display
- Decode AAC-LC and AAC-HE to PCM format and store/play
- Decode G.711 to PCM and store/play
- Decode JPEG to 422ILE format and display
- Decode JPEG and store
- Encode and dump raw video in 422I and 420P YUV format with MPEG4 SP and H264
- Encode and dump raw YUV with JPEG
- Read/capture PCM and encode, dump with G.711
- Simultaneous Decode and store/Play/display of Video and Audio/Speech streams
- Simultaneous Decode and store/display/play of Image and Audio/Speech streams
- Encode yuv to h264 and decode it back and dump the decoded output in a yuv file

**OMAP3530(WinCE)**

- Decode MPEG2, MPEG4 and H264 to YUV 420P/422ILE format and store
- Read, Encode and dump raw video in 422I and 420P YUV format with MPEG4 SP and H264

## 6 Fixed In This Release

Reference	Description

## 7 Known Issues in DVTB

Identifier	Headline	Workaround
<b>SDOCM00052712</b>	DVTB does not remove spurious data from the non standard MP3 streams	None
<b>SDOCM00055930(duplicate of 52712)</b>	DVTB does not remove spurious data from the non standard MP3 streams	None
<b>SDOCM00056787</b>	Frame rate is not being met when running multiple codec instance use cases in dvtb	None
<b>SDOCM00056425</b>	DVTB decode plays files too slow when output is via component	None
<b>SDOCM00041259</b>	OMAP3530: Control-C causes stack-trace to be dumped on to	Not supported in DVTB

	terminal in case of DVTB decode.	
<b>SDOCM00052712</b>	OMAP3530: DVTB does not remove spurious data from the non standard MP3 streams	Not supported in DVTB
<b>SDOCM00064970</b>	Omap3530: capture and encode fails in Memory_getBufferPhysicalAddress	Not supported in DVTB

## 8 DVTB Limitation

DVTB does not support evaluation combo in this release. User needs to modify the DVTB configuration file of respective Platform. For more detail, Please refer to section 2.3 of DVTB user guide.

### **DM365:**

- 1) DVTB may not be executing in real-time.
- 2) DVTB do not perform any i/p data format conversion for file based operations with an exception to JPEGENC1 class
- 3) DVTB do not perform any o/p data format conversion for file based operations for IMGDEC1 based classes
- 4) DVTB do not perform any o/p data format conversion for file based operations for VIDDEC2 based classes when 'rotation' functionality is enabled.
- 5) File based area encoding is not supported in DVTB.
- 6) DVTB does not support inputWidth and inputHeight as non multiple of 16 for mpeg4 encoder.

### **OMAP3530:**

- 1) DVTB does not support following parameters

<b>Handle</b>	<b>Parameters</b>
videnc1	forceFrame
viddec2	newFrameFlag, mbDataFlag
jpegdec1	progDisplay
h264enc1	forceFrame, mvDataEnable, streamFormat
mpeg4spenc1	forceFrame, mvDataEnable, ResyncDataEnable
mpeg2dec2	dyna_chroma_format, getDisplayHdrInfo, reverse_play
mpeg4spdec2	newFrameFlag, mbDataFlag
h264dec2	Sei_Vui_parse_flag, numNALunits

---

**DM3730:**

- 1) DVTB does not support following parameters

<b>Handle</b>	<b>Parameters</b>
videnc1	forceFrame
viddec2	newFrameFlag, mbDataFlag
jpegdec1	progDisplay
h264enc1	forceFrame, mvDataEnable, streamFormat
mpeg4spenc1	forceFrame, mvDataEnable.
mpeg2dec2	dyna_chroma_format, getDisplayHdrInfo, reverse_play
mpeg4spdec2	newFrameFlag, mbDataFlag

**DM6467:**

- 1) Output image of jpeg decoder is embedded in actual image resolution when it is used in sub Region decoding mode. For this case, Generated YUV from DVTB has width as “scaled image width” and height as “outputHeight”.

## 9 Known Issues in dependent components

### DM355

Identifier	Headline	Workaround
<b>SDOCM00052209</b>	DM355-dvtb. Board during Video+Speech Decode operations	None
<b>SDOCM00053399</b>	DM355 DVTB Goes into infinite loop during jpeg encode operation	None
<b>SDOCM00054617</b>	DVTB DM355. Area decode does not work for image decode operations using JPEG extended parameters	None

### DM6446

Identifier	Headline	Workaround
	"audenc" class cannot be run using any of the XDM 0.9 Audio Encoders due to the known limitation that no XDM 0.9 Audio Encoder will work with base params.	None

### DM6467

Identifier	Headline	Workaround
<b>SDOCM00064754</b>	Mpeg2 decoder dumps corrupted UV component for 352x240 and 320x240 resolution	None

### DM365

Identifier	Headline	Workaround
<b>SDOCM00064479</b>	Ioctl call to VIDIOC_S_COFST fails	None
<b>SDOCM00064501</b>	Image displayed on device is shifted	None

### OMAP3530

Identifier	Headline	Workaround
<b>SDOCM00070854</b>	OMAP3530: Error messages are coming while encoding in 720x576 resolution	None

## **10 Limitation in dependent components**

Following are the bottle necks in providing generic scripts for DVTB classes across platforms.

### **DM6446:**

- 1) MPEG4 encoder does not support resolution 1280x720 (DMAI default)

### **DM355:**

- 1) MPEG4 decoder does not support forceChromaFormat as XDM\_YUV\_420P (DMAI default)
- 2) MPEG4 encoder does not support reconChromaFormat as XDM\_CHROMA\_NA(DMAI default)
- 3) MPEG4 encoder maxHeight and maxWidth needs to be same as inputHeight and inputWidth respectively.
- 4) JPEG Encoder does not support maxHeight, maxWidth, inputHeight and inputWidth as 0 (DMAI default)