OpenMAX Test Bench (OMTB)

Version 01.00.01.07

CLI User Guide

Sep 27, 2011

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
Low Power Wireless	www.ti.com/lpw	Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments

Post Office Box 655303 Dallas, Texas 75265

Contents

Read This First

About This Manual

This manual is a reference guide for using the “OMTB command-line user application”. This manual depicts the user commands in ***bold red italics*** and screen outputs in **bold blue font.**

Information about Cautions and Warnings

This book may contain cautions and warnings.

This is an example of a caution statement.

A caution statement describes a situation that could potentially damage your software or equipment.

CAUTION

This is an example of a warning statement.

A warning statement describes a situation that could potentially cause harm to you.

WARNING

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

Related Documentation

- None

Revision History

Version	Date	Revision History
01.00.00.01	April 27, 2011	Support for 1) Capture -> Deinterlace -> Encode -> FileIO FileIO -> Decode -> Scale -> Display 2) Audio capture -> G.711 Encode -> G.711 Decode -> Audio Playback
01.00.00.03	July 29, 2011	Support for 1)Mpeg2 Decoder component 2)File read, file write for Video Capture, Deinterlace, Scale, video Display components 3)Sequential execution of end to end usecase i.e termination of OMTB
01.00.01.00	Aug 04, 2011	Support for 1)Cmd line mode and Scripting mode 2)Modified the make files to link the omx libraries
01.00.01.01	Aug 09, 2011	1)Updated the oms scripts to run Netra BM demo
01.00.01.02	Aug 10, 2001	Support for 1)Centaurus platform
01.00.01.03	Aug 11, 2001	Support for 1)Backward compatibility to run omtb script “./omtb-bin <script name>”
01.00.01.04	Aug 26, 2001	Bug fixes SDOCM00083667, SDOCM00083666, SDOCM00083525, SDOCM00083518, SDOCM00083669
01.00.01.05	Aug 28, 2011	Modified omtb make files, removed linking of omx obj files
01.00.01.06	Sep13, 2011	Support for Audio decode AAC component
01.00.01.07	Sep 27, 2011	1)Modified the make command to build omtb 2)Support for MP3 Decoder component 3)Support for history commands when omtb is executed in cmd line mode

Abbreviations

Table 1-1. Table of Abbreviations

Abbreviation	Description
API	Application Programming Interface
CLI	Command Line Interface
DEI	De-interlacer
DSP	Digital Signal Processor
GPP	General Purpose Processor
HDVPSS	High Definition Video Processing Sub-system
NF	Noise Filter
OMTB	OpenMax (IL Component) Test Bench
OMS	OMTB Script
OMX	OpenMax
SC	Scalar
VDEC	Video Decoder
VENC	Video Encoder
VFCC	Video Frame Capture Component
VFDC	Video Frame Display Component
VFPC	Video Frame Processing Component

THIS PAGE LEFT INTENTIONALLY BLANK

Chapter 1

OMTB Application

This chapter describes about OpenMax Test Bench (OMTB) application, it's build procedure and usage. In addition, this chapter gives a summary of all the commands supported by OMTB application, which is used to test OpenMax IL component.

1.1 Introduction

1.1.1 What is OMTB?

OMTB *or* OpenMAX Test Bench is a software application that can be used to create and connect instances of TI supported Multi-media Software Modules*, configure them and run them together as an intended use-case

It is packaged with the TI DaVinci™ Software Development Kit (SDK)

1.1.1.1 Key Features

- Scripting capability allows user to create use-cases and vary the configuration/connection as intended at run-time
- Command Line interface to create use-cases
- It supports validation of
  - ✓ Standalone OMX Component Functionality
  - ✓ OMX APIs and all possible component configurations
  - ✓ OMX Data Interfaces
    - Standard Non Tunneling (SNT)
  - ✓ Multiple instances of one or more OMX components
  - ✓ Usecases that tunnel two or more OMX components
  - ✓ Stress/Stability of OMX components and Usecases
  - ✓ Performance of OMX components and Usecases
  - ✓ Runs on Cortex - A8
  - ✓ Supports API based or Usecase based execution
  - ✓ Supports scripting mode of execution [*.oms – OMTB Scripts are supported]
  - ✓ Supports sophisticated command line help
  - ✓ Supports configuration templates
  - ✓ Allows storing/loading of pre-configured parameters from/to file enabling ease of use
  - ✓ Thin framework enables easy isolation of issues

1.1.1.2 Out of the Box Experience

Example scripts cover basic creation, configuration and connection of TI software modules which can be run straight on the Evaluation Module (EVM).

1.1.1.3 Support

Documents: Quick Start Guide, User Guide and Release Notes

- How to download, install, build and run
- How to create, configure and connect module instances

Support through TI E2E Forum:

External: http://e2e.ti.com/support/embedded/f/354.aspx
Internal: http://e2e.ti.com/support/embedded/int-embedded_software/f/118.aspx

1.1.1.4 Licensing

Open Sourced Software. Berkeley Software Division (BSD) licensed

http://www.opensource.org/licenses/bsd-license.php

1.1.2 When OMTB should be used?

To explore usage of TI DaVinciTM SDK Multi-media Software Modules in creating the use-cases

- Use example use-cases provided in OMTB package:
  - Capture Video → Encode → Store
  - Decode → Display
- Create new use-cases like:
  - Simple use-cases: Capture → Display, Capture → Resizer → Display
  - Complex use-cases: Capture → Encode → Decode → Display

To test TI software modules' APIs*

1.1.3 What is in the OMTB package?

- OMTB Source Code
- OMTB User Guide
- OMTB Release Notes
- Makefiles for re-building OMTB binaries
- Example Scripts that demonstrate usage (Creation, Configuration & Connection of Software modules)

1.2 List of supported components.

1) H264 video decode component
2) H264 video encode component
3) HDVPSS video frame capture component
4) HDVPSS video frame processing components – DEI, NF and SC
5) HDVPSS video frame display component
6) TVP Decoder component
7) Display Controller component
8) Audio capture and playback using ALSA drivers
9) Mpeg2 decoder component
10) AAC decoder component
11) MP3 decoder component

1.3 Build Procedure

1.3.1 DM81xx

1.3.1.1 Steps to build the executable

1. Invoke “make components” from EZSDK directory
2. Go to omtb_xx_xx_xx_xx directory
3. Run the following make command

   *“make dm81xxbm **ROOTDIR**=<OMTB INSTALL DIR> **EZSDK_INSTALL_DIR**=<SDK INSTALL DIR> **DEST_ROOT**=<DESTINATION DIR> **OMX_INSTALL_DIR**=<OMX INSTALL DIR>/packages **fc_PATH**=<FC INSTALL DIR**> ce_PATH**=<CE INSTALL DIR> **osal_PATH**=<OSAL INSTALL DIR> **linuxutils_PATH**=<LINUXUTILS INSTALL DIR> **ipc_PATH**=<IPC INSTALL DIR> **syslink_PATH**=<SYSLINK INSTALL DIR> **xdc_PATH**=<XDC INSTALL DIR> **lindevkit_PATH**=<LINUX DEVKIT DIR>/arm-none-linux-gnueabi/usr **CODEGEN_PATH_A8**=<CSTOOL DIR> **uia_PATH**=<UIA INSTALL DIR> **PLATFORM**=ti816x-evm **or** ti814x-evm”*

4. omtb_dm81xxbm_a8host.xv5T will be created under the following platform specific directory

   If ***PLATFORM**=ti816x-evm*

   <DEST_ROOT >*/dm816xbm/bin/ti816x-evm*

   If ***PLATFORM**=ti814x-evm*

   <DEST_ROOT >*/dm814xbm/bin/ti814x-evm*

5. Use “*make clean_dm81xxbm **ROOTDIR**=<OMTB ROOT DIR> **CODEGEN_PATH_A8**=<CSTOOL DIR>”* to clean all the files.

1.3.1.2 Steps to run the executable

1. Create a folder in the target file system say <target_fs>/home/omtb_01_00_01_07
2. Copy the following into the target folder:
   - omtb_dm81xxbm_a8host.xv5T
   - dm816x_hdvicp.xem3 **or** dm814x_hdvicp.xem3
   - dm816x_hdvpss.xem3 **or** dm814x_hdvpss.xem3
   - syslink.ko
   - ti81xxhdmi.ko
   - prcm_config_app
   - firmware_loader

- vpss.ko

3. Power on the board, wait for the shell prompt.
4. Run the following commands
   - ./ prcm_config_app s
   - insmod syslink.ko
   - ./firmware_loader 1 dm816x_hdvicp.xem3 start <path mem map file>

     or

     ./firmware_loader 1 dm814x_hdvicp.xem3 start <path mem map file>
   - ./firmware_loader 2 dm816x_hdvpss.xem3 start <path mem map file>

     or

     ./firmware_loader 2 dm814x_hdvpss.xem3 start <path mem map file>
   - insmod vpss.ko mode=hdmi:1080p-60,hdcomp:1080p-60 i2c_mode=1

     Note: In the above command for DM816x set i2c_mode=1, for DM814x set i2c_mode=0
   - insmod ti81xxhdmi.ko

Note: All the run commands mentioned in point 4 can be ignored if they are called as part of board initialization.

- ./omtb_dm81xxbm_a8host.xv5T -s <OMTB script>

  or

  ./omtb_dm81xxbm_a8host.xv5T <OMTB script>

1.4 OMTB Configuration Template

OMX component can be configured run time using application's configuration parameter template. Template is the set of configuration parameters which user can set or view for OMX component configuration. Templates are application's local copy of configuration parameters from which data will be copied to components configuration structure or application's private data structure. Template memory is allocated dynamically based on the template number and template is already allocated or not. OMTB also provides commands for storing and loading configuration templates to/from file, reset the template configuration, add/remove templates and set/get template parameters. Thus template # is the index to application's configuration parameter set stored into memory. Maximum template is limited to 10 in current release. So user can use 0 to 9 as template # based on their configuration requirements.

1.5 OMTB Instance Number

OMTB instance manager will support multiple instances of single OMX component or different OMX components. OMTB will identify the particular instance by the component instance number. OMTB will internally handle the OMX component handles based on the component class/type to store and fetch it. Thus instance # is the index for the particular OMX component instance which will be used by the OMTB to handle that particular instance. User needs to pass instance number only and OMTB will map it to particular handle pointer from the OMTB instance management structure. Maximum instance number is limited to 16 in current release. So user can use 0 to 15 as the instance number based on his convenience.

1.6 Basic Sequence of Commands

Following is the basic execution sequence to test OpenMax components using OMTB:

1. Initializing the OpenMax core.
2. Get the OpenMax component handle.
3. Set the OpenMax component parameters if any.
4. Apply command to change the component state to IDLE (Ready).
5. Apply command to change the component state to EXEC (Executing).
6. Apply command to change the component state to IDLE (Ready).
7. Apply command to change the component state to LOADED (Release the allocated resources).
8. Free the OpenMax component handle.
9. Un-initializing the OpenMax core.

Usage of OMTB commands:

To get a list of valid OMTB commands, type:

```
OMTB> omx
```

To invoke any OMTB command, type:

```
OMTB> omx <command> <command_params>
```

If the command interpreter finds an error in the command line, it prints the correct syntax of the command up to the level where the error occurred and provides valid options in that juncture.

The error handling of the command interpreter can also be used as a learning tool to find the correct syntax of a command, even without knowing the command variable.

For example:

```
OMTB> omx func
```

Prints the following output:

```
omx func viddec <params>
omx func videnc <params>
```

The output indicates that the `omx func` command takes `viddec` parameter or `videnc` parameter followed by `<h264vdec | h264venc>` `<template #>` and `<codec_name>`. Current release only supports one video decoder and one video encoder components so multiple codec options will not be displayed.

Note: OMTB commands are case-sensitive**.** Command responses given in the user guide might not exactly match the actual responses**.**

1.7 Invoking OMTB Application

OMTB application can be used in

1. **Script mode**

Script has to be passed as an argument to the OMTB, the command execution will be started from the script until each command in the script executed and there response will be displayed.

E.g. If script.oms file contains "**`omx setp 0 h264vdec infile test.264`**" command then following will be displayed:

```
OMTB> omx setp 0 h264vdec infile test.264
Ok
```

1.8 OMTB Commands

Following are the OMTB command groups:

- **OMTB OMX Commands**
  - api - OMX component api management commands

    This set of commands allows user to initialize, execute (state transitions & configurations) and un-initialize the OpenMax component.
  - func command

    This command allows user to execute decode in one go
  - apitest - OMX component api test commands

This set of commands allows user to validate OMX APIs for different components.

- **OMTB Utility Commands**
  - -s command

    This command will provide the option to run the command from the oms script file runtime without exiting application. OMTB command prompt will come once end of file is reached.

  - setp and getp commands

    This set of commands allows user to set and retrive the configuration parameters of the OpenMax component.

  - omtb_rel_info

    This command will display the current OMTB version.

- add and remove OMTB configuration template

  This command will add/remove the OMTB configuration template to/from memory if they are not already added/removed based on the template number passed.

- store and load OMTB configuration template to/from file

  This command will allow user to store the current template configuration to file or load the previously stored configuration from file. Command will add the template if specified template is not already allocated.

- reset OMTB configuration template

  This command will allow user to reset the configuration present in template. Reset will set all the numbers to zero and all the strings to NULL.

- tog_arm_load command

  This command allows user to toggle the ARM CPU load display.

- comp_dbg_log command

  This command allows user to toggle the OMX Component debug log display.

- omtb_dbg_lvl command

  This command allows user to set the OMTB debug log display level.

To see all the available OMTB commands just type omx as follows:

```
OMTB> omx
OMTB Commands:
---------------
   func                 api                  api_test
```

```
Utility Commands:
------------------
   -s                  setp                getp
   omtb_rel_info       add                 remove
   load                store               reset
   tog_arm_load        comp_dbg_log        omtb_dbg_lvl
```

1.8.1 OMTB OMX Commands

1.8.1.1 api - OpenMax Component API Management Commands

1.8.1.1.1 Initialize the OpenMax core

‖ Command Syntax

```
Omx api init
```

‖ Function

This command initializes OpenMax core.

‖ Constraints

OpenMax core initialization is common between different OpenMax components. So this command should be used once before any other commands.

Only after user initializes the OpenMax core, user can start using the other functionalities of OpenMax component.

‖ Example

```
OMTB> omx api init
OMX_ErrorNone
OK
```

1.8.1.1.2 Get the OpenMax component handle

‖ Command Syntax

```
omx api gethandle <component> <instance #> <template #>
```

‖ Function

This command creates the OpenMax component instance and returns the handle for the component.

‖ Constraints

OpenMax core initialization must be done before invoking this command. Also valid parameters must be set before invoking this command using "`omx setp <template #> <component> <params>`".

Only after user gets the OpenMax component handle, user will be able to execute other OpenMax component related commands.

‖ Example

```
OMTB> omx api gethandle h264vdec 0 0
OMTB-H264vdec Instance #: 0
OK
```

1.8.1.1.3 Get the OpenMax component version

‖ Command Syntax

```
omx api compver <component> <instance #>
```

‖ Function

This command gives the OpenMax component version and specification version against which component is developed.

‖ Constraints

OpenMax core initialization and OpenMax component get handle commands must be invoked before invoking this command using "`Omx api init`" and "`omx api gethandle <component> <instance #> <template #>` respectively.

‖ Example

```
OMTB> omx api compver h264vdec 0
Component Version is:1.1.0
Spec Version is:     101
OK
```

1.8.1.1.4 Set the OpenMax component parameters

‖ Command Syntax

```
omx api setparam <component> <instance #> <template #>
<port #> <OMX Index>
```

‖ Function

This command allows user to set the OpenMax component parameter.

‖ Constraints

OpenMax core initialization and OpenMax component get handle commands must be invoked before invoking this command using "`Omx api init`" and "`omx api gethandle <component> <instance #> <template #>` respectively.

‖ Example

```
OMTB> omx api setparam h264venc 0 0 1
OMX_IndexParamVideoBitrate
OMX_ErrorNone
OK
```

1.8.1.1.5 Get the OpenMax component parameters

‖ Command Syntax

```
omx api getparam <component> <instance #> <template #>
<port #> <OMX Index>
```

‖ Function

This command allows user to get the OpenMax component parameter.

‖ Constraints

OpenMax core initialization and OpenMax component get handle commands must be invoked before invoking this command using "`Omx api init`" and "`omx api gethandle <component> <instance #> <template #>` respectively.

‖ Example

```
OMTB>  omx  api  getparam  h264venc  0  0  1
OMX_IndexParamVideoBitrate
```

1.8.1.1.6 Set the OpenMax component parameters runtime

‖ Command Syntax

```
omx api setconfig <component> <instance #> <template #>
<port #> <OMX Index>
```

‖ Function

This command allows user to set the OpenMax component parameter in all the state except invalid.

‖ Constraints

OpenMax core initialization and OpenMax component get handle commands must be invoked before invoking this command using "`Omx api init`" and "`omx api gethandle <component> <instance #> <template #>`" respectively.

‖ Example

```
OMTB>  omx  api  setconfig  h264venc  0  0  1
OMX_IndexConfigVideoBitrate
OMX_ErrorNone
OK
```

1.8.1.1.7 Get the OpenMax component parameters runtime

‖ Command Syntax

```
omx api getconfig <component> <instance #> <template #>
<port #> <OMX Index>
```

‖ Function

This command allows user to get the OpenMax component parameter in all the state except invalid.

‖ Constraints

OpenMax core initialization and OpenMax component get handle commands must be invoked before invoking this command using "`Omx api init`" and "`omx api gethandle <component> <instance #> <template #>`" respectively.

‖ Example

```
OMTB>  omx  api  getparam  h264venc  0  0  1
OMX_IndexConfigVideoBitrate
```

1.8.1.1.8 Get the OpenMax component state

‖ Command Syntax

```
omx api getstate <component> <instance #>
```

‖ Function

This command allows user to get the OpenMax component current state.

‖ Constraints

OpenMax core initialization and OpenMax component get handle commands must be invoked before invoking this command using "Omx api init" and "omx api gethandle <component> <instance #> <template #>" respectively.

‖ Example

```
OMTB> omx api getstate h264vdec 0
OMTB-OMX_ErrorNone
OMTB-<component, instance , state> ==> <h264vdec, 0 , OMX_StateIdle>
OK
```

1.8.1.1.9 Get the OpenMax component extension index

‖ Command Syntax

```
omx api getextindex <component> <instance #> <Index String>
```

‖ Function

This command allows user to get the OMX standard structure index extended from OMX standard definitions.

[Note] Currently OMX components don't support it so only support is added but not tested.

‖ Constraints

OpenMax core initialization and OpenMax component get handle commands must be invoked before invoking this command using "Omx api init" and "omx api gethandle <component> <instance #> <template #>" respectively.

‖ Example

```
OMTB> omx api getextindex h264vdec 0 h264_TI_CfgParams
```

Note: This functionality is not supported in the component currently.

1.8.1.1.10 Send the commands to OpenMax component

‖ Command Syntax

```
omx api sendcommand <params>
```

<***Following message will be displayed on entering "omx api sendcommand"***>

```
omx api sendcommand state <component> <instance #> <loaded | idle | exec | pause | waitforresources>
```

‖ Function

This command allows user to change the OpenMax component state, port definition, flushing the buffer of the particular port and mark the particular buffer.

[Note] Currently only state transition loaded, idle and exec is fully tested and mark buffer is not tested.

‖ Constraints

OpenMax core initialization and OpenMax component get handle commands must be invoked before invoking this command using "Omx api

init" and "omx api gethandle <component> <instance #> <template #>" respectively.

‖ Example

```
OMTB> omx api sendcommand state h264vdec 0 idle
OMX_ErrorNone
OK
```

1.8.1.1.11 Connect the OpenMax components

‖ Command Syntax

```
omx api connect <params>
```

‖ Function

This command allows user to pass the information like components that are connected to each other, OMTB will use this information to transfer the data buffers to the components.

‖ Constraints

OpenMax core initialization and OpenMax component get handle commands must be invoked before invoking this command using "Omx api init" and "omx api gethandle <component> <instance #> <template #>" respectively.

‖ Example

```
OMTB> omx api connect h264venc 0 1 h264vdec 0 0
OMX_ErrorNone
OK
```

1.8.1.1.12 Free the OpenMax component handle

‖ Command Syntax

```
omx api freehandle <component> <instance #>
```

‖ Function

This command releases the OpenMax component handle by deleting the instance.

‖ Constraints

OpenMax core initialization and OpenMax get component handle must be invoked before invoking this command using"Omx api init" and "omx api gethandle <component> <instance #> <template #>" respectively.

‖ Example

```
OMTB> omx api freehandle h264vdec 0
OMX_ErrorNone
OMTB-Deleted H264Vdec Instance #: 0
OK
```

1.8.1.1.13 Get the supported component list

‖ Command Syntax

```
omx api getcompname
```

‖ Function

This command displays list of all the supported components.

‖ Constraints

OpenMax core initialization should be done before invoking this API.

‖ Example

```
OMTB> omx api getcompname
OMTB-Component 0: OMX.TI.DUCATI1.VIDEO.H264D
OMTB-Component 1: OMX.TI.DUCATI1.VIDEO.H264E
OMTB-OMX_ErrorNoMore
OK
```

1.8.1.1.14Get the roles by component

‖ Command Syntax

```
omx api rolesbycomp <component>
```

‖ Function

This command displays roles supported by the specified component.

‖ Constraints

OpenMax core initialization should be done before invoking this API.

‖ Example

```
OMTB> omx api rolesbycomp OMX.TI.DUCATI1.VIDEO.H264D
OMTB-<component , # of roles> :
<OMX.TI.DUCATI1.VIDEO.H264D , 0>
OMTB-The Number or roles is 0
The component selected is not correct for the purpose of
this test
OMTB-OMX_ErrorNone
OK
```

1.8.1.1.15Get the components by role

‖ Command Syntax

```
omx api compsbyrole <Role name>
```

‖ Function

This command displays the components supporting specified role.

‖ Constraints

OpenMax core initialization should be done before invoking this API.

‖ Example

```
OMTB> omx api compsbyrole VIDEO

OMTB-Number of components per role <VIDEO> : 0
OK
```

1.8.1.1.16Uninitialize the OpenMax core

‖ Command Syntax

```
Omx api uninit
```

‖ Function

This command un-initializes the OpenMax core.

‖ Constraints

OpenMax core must be initialized before invoking "`Omx api init`" command. User must un-initialize the OpenMax core exiting from OpenMax component or OMTB application. This command should be invoked at the end of any other commands.

‖ Example

```
OMTB> omx api uninit
OMX_ErrorNone
OK
```

1.8.2 OMTB Utility Commands

1.8.2.1 Run oms script using –s command runtime

‖ Command Syntax

```
omx -s <script_file_name>
```

‖ Function

This command runs the command from the OMS script file run time till the end of file is reached.

‖ Constraints

This command reads the data line by line from the file so no of characters in one line should not increase beyond 200 characters (Maximum command line input length allowed by OMTB).

‖ Example

```
OMTB> omx -s script.oms
OK
```

1.8.2.2 Set the template parameter

‖ Command Syntax

```
omx setp <template #> <component> <params>
```

‖ Function

This command sets the OMTB configuration template parameters used for component configuration.

‖ Constraints

None

‖ Example

```
OMTB> omx setp 0 h264vdec frame_width 176
OK
                or

OMTB> omx setp 0 h264venc OMX_IndexParamVideoBitrate
nTargetBitrate 4000000

OK
```

1.8.2.3 Get the template parameter

‖ Command Syntax

```
omx getp <template #> <component> <params>
```

‖ Function

This command sets the components template parameter used for configuration.

‖ Constraints

None

‖ Example

```
OMTB> omx getp 0 h264vdec frame_width
Template 0:

 H264 Video Decoder Config Parameters:

   Input Frame Width is..... 176
OK

OMTB> omx getp 0 h264vdec

Template 0:


H264 Video Decoder Config Parameters:


   OMX Component name
is.....OMX.TI.VIDEO.H264D

   Input File name
is.....h264_sample.264

   Output File name
is.....h264_test.yuv

   Frame Size File name
is.....frame_data.txt

   Input Frame Width is.....176

   Input Frame Height is.....144

   O/P Chroma Format.....420P

   Data O/P Mode.....file

   Buffer allocation Mode.....use

   Buffer allocation Mode.....use

   Number of i/p buffer is.....1
   Number of o/p buffer is.....1
OK
```

1.8.2.4 OMTB release information display

‖ Command Syntax

```
omx omtb_rel_info
```

‖ Function

This command displays OMTB release version number.

‖ Constraints

None

‖ Example

```
OMTB> omx omtb_rel_info
OMTB-OMTB - OMTB Version: OMTB 00.00.00.04
OK
```

1.8.2.5 Add OMTB configuration template

‖ Command Syntax

```
omx add <template #>
```

‖ Function

This command adds OMTB configuration template into memory.

‖ Constraints

None

‖ Example

```
OMTB> omx add 1
OK
```

1.8.2.6 Remove OMTB configuration template

‖ Command Syntax

```
omx remove <template #>
```

‖ Function

This command removes OMTB configuration template from the memory.

‖ Constraints

None

‖ Example

```
OMTB> omx remove 1
OK
```

1.8.2.7 Load OMTB configuration template

‖ Command Syntax

```
omx load <template #> <cfg_file_name>
```

‖ Function

This command loads specified OMTB configuration template from the file.

‖ Constraints

OMTB internally adds the "`omx setp <template #>`" before each of the command read from the configuration file. So "`omx setp <template #>`" should not be added to the commands while creating configuration file manually. OMTB store command "`omx store <template #> <cfg_file_name>`" takes care of this while storing the data to file.

‖ Example

```
OMTB> omx load 1 cfg.cfg

OMTB> omx setp 0 h264vdec frame_width 176
OK
```

1.8.2.8 Store OMTB configuration template

‖ Command Syntax

```
omx store <template #> <cfg_file_name>
```

‖ Function

This command stores specified OMTB configuration template to the file.

‖ Constraints

None

‖ Example

```
OMTB> omx store 1 cfg.cfg
OK
```

1.8.2.9 Reset OMTB configuration template

‖ Command Syntax

```
omx reset <template #> <component> <params>
```

‖ Function

This command resets specified OMTB configuration template. Reset will set all the integer params to zero and strings to NULL.

‖ Constraints

None

‖ Example

```
OMTB> omx reset 0 h264vdec infile
OK
```

1.8.2.10 Toggle ARM CPU load display

‖ Command Syntax

```
omx tog_arm_load
```

‖ Function

This command toggles the ARM CPU load display.

‖ Constraints

None

‖ Example

```
OMTB> omx tog_arm_load
OK
```

1.8.2.11 Toggle OMX component debug log display

‖ Command Syntax

```
omx comp_dbg_log
```

‖ Function

This command toggles the OMX component debug log display.

‖ Constraints

None

‖ Example

```
OMTB> omx comp_dbg_log
OK
```

1.8.2.12 Set OMTB debug log display level

‖ Command Syntax

```
omx omtb_dbg_lvl <debug level>

where <debug level> can be, 0x00 - Disable Logs;
```

0x02 - Simple Logs;

0x04 - Default Logs;

0x08 - Function Names;

0x10 - Debug Logs;

0x80 - Compulsary Logs/Error Logs

‖ Function

This command sets the OMTB debug log level display.

‖ Constraints

None

‖ Example

```
OMTB> omx omtb_dbg_lvl 0x00
OK
```

1.8.3 OMTB Template Configuration Commands

OMTB uses configuration templates as a database to store the configuration parameters for OpenMax component. These configuration parameters are used to initialize OpenMax component parameters; In addition, configuration parameters are used to create and set the port configuration for OpenMax component instances. The configuration parameters must be set correctly before invoking appropriate commands.

OMTB application allows multiple configuration templates to be added and removed. OMTB also allows storing/loading the template configuration to/from the file. OMTB utility commands provide facility to process the template database.

1.8.3.1 Show template values

‖ Command Syntax

`omx getp <template #> <component> <params>`

‖ Function

This command shows the current values set in the given template for the given module.

Module and module-params are optional. If user does not pass the module name (`h264vdec`) and module parameters (`params`), all the configuration parameters in the template are displayed.

‖ Example

```
OMTB> omx getp 0 h264vdec frame_width
Template 0:

 H264 Video Decoder Config Parameters:

   Input Frame Width is.......................... 176
OK

OMTB> omx getp 0 h264vdec
Template 0:
```

```
 H264 Video Decoder Config Parameters:

   OMX Component name
is.....................OMX.TI.VIDEO.H264D

   Input File name
is.....................h264_sample.264

   Output File name
is.....................h264_test.yuv

   Frame Size File name
is.....................frame_data.txt

   Input Frame Width is.....................176

   Input Frame Height is.....................144

   O/P Chroma Format.....................420P

   Data O/P Mode.....................file

   Buffer allocation Mode.....................use

   Buffer allocation Mode.....................use

   Number of i/p buffer is.....................1
   Number of o/p buffer is.....................1
OK
```

1.8.3.2 Set template values

‖ Command Syntax

```
omx setp <template #> <component> <index> <params>
```

‖ Function

This command sets new value for the given parameter of a specific module.

‖ Example

```
OMTB> omx setp 0 h264vdec frame_width 176
OK
```

‖ Constraints

Each individual parameter type and size is constraint. User must be aware of allowed maximum limit for all the parameters. Command entered from the command line must not exceed 255 bytes including NULL character.

1.8.3.3 Load OMTB configuration template

‖ Command Syntax

```
omx load <template #> <cfg_file_name>
```

‖ Function

This command loads specified OMTB configuration template from the file.

‖ Constraints

OMTB internally adds the "`omx setp <template #>`" before each of the command read from the configuration file. So "`omx setp <template #>`"

should not be added to the commands while creating configuration file manually. OMTB store command "`omx store <template #> <cfg_file_name>`" takes care of this while storing the data to file.

‖ Example

```
OMTB> omx load 1 cfg.cfg

OMTB> omx setp 0 h264vdec frame_width 176
OK
OK
```

1.8.3.4 Store OMTB configuration template

‖ Command Syntax

```
omx store <template #> <cfg_file_name>
```

‖ Function

This command stores specified OMTB configuration template to the file.

‖ Constraints

None

‖ Example

```
OMTB> omx store 1 cfg.cfg

OK
```

1.8.3.5 Reset OMTB configuration template

‖ Command Syntax

```
omx reset <template #> <component> <params>
```

‖ Function

This command resets the current values set in the given template for the given module.

Module and module-params are optional. If user does not pass the module name (`h264vdec`) and module parameters (`params`), all the configuration parameters in the template are reset. All the integer params will be set to zero and strings to NULL.

‖ Example

```
OMTB> omx reset 0 h264vdec frame_width
OK
```

‖ Constraints

None.

THIS PAGE LEFT INTENTIONALLY BLANK

Troubleshooting

A.1 None

None

A.2 None

None