Ambarella

SDK6 API Middleware (Flow)

Version 1.2
June 24, 2015

Confidentiality Notice:

The contents of this document are proprietary and confidential information of Ambarella Inc.

The material in this document is for information only. Ambarella assumes no responsibility for errors or omissions and reserves the right to change, without notice, product specifications, operating characteristics, packaging, ordering, etc. Ambarella assumes no liability for damage resulting from the use of information contained in this document. All brands, product names and company names are trademarks of their respective owners.

US
3101 Jay Street
Ste. 110
Santa Clara, CA 95051, USA
Phone: +1.408.734.8888
Fax: +1.408.734.0788

Hong Kong
Unit A&B, 18/F, Spectrum Tower
53 Hung To Road
Kwun Tong, Kowloon
Phone: +85.2.2806.8711
Fax: +85.2.2806.8722

Korea
6 Floor, Hanwon-Bldg.
Sunae-Dong, 6-1, Bundang-Gu
SeongNam-City, Kyunggi-Do
Republic of Korea 463-825
Phone: +031.717.2780
Fax: +031.717.2782

China - Shanghai
9th Floor, Park Center
1088 Fangdian Road, Pudong New District
Shanghai 201204, China
Phone: +86.21.6088.0608
Fax: +86.21.6088.0366

Taiwan
Suite C1, No. 1, Li-Hsin Road 1
Science-Based Industrial Park
Hsinchu 30078, Taiwan
Phone: +886.3.666.8828
Fax: +886.3.666.1282

Japan - Yokohama
Shin-Yokohama Business Center Bldg. 5th Floor
3-2-6 Shin-Yokohama, Kohoku-ku,
Yokohama, Kanagawa, 222-0033, Japan
Phone: +81.45.548.6150
Fax: +81.45.548.6151

China - Shenzhen
Unit E, 5th Floor
No. 2 Finance Base
8 Ke Fa Road
Shenzhen, 518057, China
Phone: +86.755.3301.0366
Fax: +86.755.3301.0966

I Contents

1 Display

1.1 Display: Overview

This chapter provides information regarding the display module which is used to control an output device. In most cases, two video output channels are provided: a digital-only channel as well as a fully functional channel.

1.2 Display: List of Functions

- (Section 1.2.1) AmpDisplay_Create
- (Section 1.2.2) AmpDisplay_CreateWindow
- (Section 1.2.3) AmpDisplay_Delete
- (Section 1.2.4) AmpDisplay_DeleteWindow
- (Section 1.2.5) AmpDisplay_GetDefaultCfg
- (Section 1.2.6) AmpDisplay_GetDefaultInitCfg
- (Section 1.2.7) AmpDisplay_GetDefaultWindowCfg
- (Section 1.2.8) AmpDisplay_GetDeviceCfg
- (Section 1.2.9) AmpDisplay_GetInfo
- (Section 1.2.10) AmpDisplay_GetWindowCfg
- (Section 1.2.11) AmpDisplay_Init
- (Section 1.2.12) AmpDisplay_SetDeviceCfg
- (Section 1.2.13) AmpDisplay_SetWindowActivateFlag
- (Section 1.2.14) AmpDisplay_SetWindowCfg
- (Section 1.2.15) AmpDisplay_Start
- (Section 1.2.16) AmpDisplay_Stop
- (Section 1.2.17) AmpDisplay_Update
- (Section 1.2.18) AmpDisplay_SetMaxVoutSize

1.2.1 AmpDisplay_Create

API Syntax:

AmpDisplay_Create (AMP_DISP_CFG_s * Cfg, AMP_DISP_HDLR_s ** DispHdlr)

Function Description:

- This function is used to create a display handler, which manages window handler memory.

Parameters:

Type	Parameter	Description
AMP_DISP_CFG_s *	**Cfg**	Display handler configuration parameters (Section 1.2.1.1)
AMP_DISP_HDLR_s **	**DispHdlr**	The display handler instance for the user (Section 1.2.1.4)

*Table 1-1. Parameters for SDK6 API Middleware Display API **AmpDisplay_Create()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-2. Returns for SDK6 Middleware Display API **AmpDisplay_Create()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.1.1 AmpDisplay_Create > AMP_DISP_CFG_s

Type	Field	Description
AMP_DISP_DEV_CFG_s	**Device**	Device information (Section 1.2.1.2)
UINT32	**MaxNumWindow**	Maximum number of window handlers set from upper-layer VOUT system type
AMP_DISP_VOUT_SYSTEM_e	**SystemType**	Device system type AMP_DISP_NTSC: NTSC AMP_DISP_PAL: PAL / SECAM
UINT32	**ScreenRotate**	Flag for device screen rotation
UINT8	**Out3dMode**	3D output mode
AMP_CALLBACK_f	**CbCfgUpdated**	Callback for VOUT change use
AMP_CALLBACK_f	**CbCodecEvent**	Callback for VOUT change use

*Table 1-3. Definition of **AMP_DISP_CFG_s** for Display API **AmpDisplay_Create()**.*

1.2.1.2 AmpDisplay_Create > AMP_DISP_DEV_CFG_s

Type	Field	Description
AMP_DISP_CHANNEL_IDX_e	**Channel**	Channel used by the device AMP_DISP_CHANNEL_DCHAN: VOUT channel 0, for digital output only AMP_DISP_CHANNEL_FCHAN: VOUT channel 1, for full-function output only AMP_DISP_CHANNEL_NUM: Total number of VOUT channels
AMP_DISP_DEV_IDX_e	**DeviceId**	Device type AMP_DISP_LCD: Display handler on LCD AMP_DISP_CVBS: Display handler on CVBS AMP_DISP_HDMI: Display handler on HDMI AMP_DISP_NONE: Display handler on Nothing
UINT32	**DeviceMode**	Resolution
UINT32	**DeviceAr**	Device aspect ratio
AMP_DISP_CUSTOM_CFG_s	**CustomCfg**	Customize CFG, set only if you do not want default value

*Table 1-4. Definition of **AMP_DISP_DEV_CFG_s** for Display API **AmpDisplay_Create()**.*

1.2.1.3 AmpDisplay_Create > AMP_DISP_CUSTOM_CFG_s

Type	Field	Description
UINT8	**EnCustomCfg**	Enable custom config for the device or not
AMBA_HDMI_VIDEO_FRAME_LAYOUT_e	**Cgf.HDMI.FrameLayout**	2D/2D video transmission

Type	Field	Description
AMBA_DSP_VOUT_HDMI_OUTPUT_MODE_e	**Cgf.HDMI.PixelFormat**	RGB/YCbCr color space
AMBA_HDMI_QUANTIZATION_RANGE_e	**Cgf.HDMI.QuantRange**	Quantization range of RGB/YCC
HDMI_AUDIO_SAMPLE_RATE_e	**Cgf.HDMI.SampleRate**	Audio sample rate
HDMI_AUDIO_CHANNEL_ALLOC_e	**Cgf.HDMI.SpeakerAlloc**	Audio channel/speaker allocation ID
HDMI_AUDIO_CLOCK_FREQUENCY_e	**Cgf.HDMI.OverSample**	Audio clock oversampling ratio
AMP_DISP_LCD_BACKLIGHT_e	**Cgf.LCD.BackLight**	LCD backlight control

*Table 1-5. Definition of **AMP_DISP_CUSTOM_CFG_s** for Display API **AmpDisplay_Create()**.*

1.2.1.4 AmpDisplay_Create > AMP_DISP_HDLR_s

Type	Field	Description
void *	**Ctx**	Pointer to codec context

*Table 1-6. Definition of **AMP_DISP_HDLR_s** for Display API **AmpDisplay_Create()**.*

1.2.2 AmpDisplay_CreateWindow

API Syntax:

AmpDisplay_CreateWindow (AMP_DISP_HDLR_s * DispHdlr, AMP_DISP_WINDOW_CFG_s * Cfg, AMP_DISP_WINDOW_HDLR_s ** WindowHdlr)

Function Description:

- This function is used to create a window handler, which manages the location of elements on the display.

Parameters:

Type	Parameter	Description
AMP_DISP_HDLR_s **	**DispHdlr**	The display handler instance for the user (Section 1.2.1.4)
AMP_DISP_WINDOW_CFG_s *	**Cfg**	Window configuration parameters (Section 1.2.2.1)
AMP_DISP_WINDOW_HDLR_s **	**WindowHdlr**	Window handler instance for the user (Section 1.2.2.5)

*Table 1-7. Parameters for SDK6 API Middleware Display API **AmpDisplay_CreateWindow()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-8. Returns for SDK6 Middleware Display API **AmpDisplay_CreateWindow()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.2.1 AmpDisplay_CreateWindow > AMP_DISP_WINDOW_CFG_s

Type	Field	Description
AMP_DISP_WINDOW_SRC_e	**Source**	Source of the window AMP_DISP_OSD: OSD window source AMP_DISP_ENC: ENC window source AMP_DISP_DEC: DEC window source AMP_DISP_DEFIMG: Default image window source AMP_DISP_BACKGROUND_COLOR: Background window source

union {
struct {
void * OsdHdlr
} Osd
struct {
AMBA_DSP_CHANNEL_ID_u VinCh
AMP_2D_BUFFER_s BlendTable
UINT8 ViewZoneID
AMP_ROTATION_e Rotate
} Enc
struct {
void * DecHdlr
} Dec
struct {
AMP_YUV_BUFFER_s * Image
UINT8 FieldRepeat
} DefImage
}
	SourceDesc	Information to fetch source.
		
		For OSD, it will be **OsdHdlr**. For ENC it will be **VinCh**. For DEC, it will be video/still **DecHdlr**.
		
		VinCh: VIN channel source
		BlendTable: Blend table used when overlapping with other ENC windows (Section 1.2.2.2)
		ViewZoneID: Indicates the view zone of the larger input; Default 0
		Rotate: Rotate the source
		Image: (Section 1.2.2.3)

AMP_AREA_s	**CropArea**	The crop area of the window source buffer (Section 1.2.2.4)
AMP_AREA_s	**TargetAreaOnPlane**	Window display position on the device (Section 1.2.2.4)
UINT32	**Layer**	A small number of layers would be placed at the bottom.

*Table 1-9. Definition of **AMP_DISP_WINDOW_CFG_s** for Display API **AmpDisplay_CreateWindow()**.*

1.2.2.2 AmpDisplay_CreateWindow > AMP_2D_BUFFER_s

Type	Field	Description
UINT8 *	**Buf**	Buffer
UINT32	**Width**	Buffer width
UINT32	**Height**	Buffer height
UINT32	**Pitch**	Buffer pitch

*Table 1-10. Definition of **AMP_2D_BUFFER_s** for Display API **AmpDisplay_CreateWindow()**.*

1.2.2.3 AmpDisplay_CreateWindow > AMP_YUV_BUFFER_s

Type	Field	Description
AMP_COLOR_FORMAT_e	**ColorFmt**	Color format of the YUV buffer
UINT32	**Width**	Buffer width
UINT32	**Height**	Buffer height
UINT32	**Pitch**	Buffer pitch
UINT8 *	**LumaAddr**	Luma address
UINT8*	**ChromaAddr**	Chroma address
AMP_AREA_s	**AOI**	Area of interest (Section 1.2.2.4)

*Table 1-11. Definition of **AMP_YUV_BUFFER_s** for Display API **AmpDisplay_CreateWindow()**.*

1.2.2.4 AmpDisplay_CreateWindow > AMP_AREA_s

Type	Field	Description
UINT32	**X**	X offset of the area
UINT32	**Y**	Y offset of the area
UINT32	**Width**	Width of the area
UINT32	**Height**	Height of the area

*Table 1-12. Definition of **AMP_AREA_s** for Display API **AmpDisplay_CreateWindow()**.*

1.2.2.5 AmpDisplay_Create > AMP_DISP_WINDOW_HDLR_s

Type	Field	Description
void *	**Ctx**	Pointer to codec context

*Table 1-13. Definition of **AMP_DISP_WINDOW_HDLR_s** for Display API **AmpDisplay_CreateWindow()**.*

1.2.3 AmpDisplay_Delete

API Syntax:

AmpDisplay_Delete (AMP_DISP_HDLR_s * DispHdlr)

Function Description:

- This function is used to delete a window handler, which clears the display handler structure.

Parameters:

Type	Parameter	Description
AMP_DISP_ HDLR_s **	**DispHdlr**	The display handler instance for the user (Section 1.2.1.4)

*Table 1-14. Parameters for SDK6 API Middleware Display API **AmpDisplay_Delete()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-15. Returns for SDK6 Middleware Display API **AmpDisplay_Delete()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.4 AmpDisplay_DeleteWindow

API Syntax:

AmpDisplay_DeleteWindow (AMP_DISP_WINDOW_HDLR_s * WindowHdlr)

Function Description:

- This function is used to delete a window handler, which clears the window handler structure.

Parameters:

Type	Parameter	Description
AMP_DISP_ WINDOW_ HDLR_s **	**WindowHdlr**	Window handler instance for the user (Section 1.2.2.5)

Table 1-16. Parameters for SDK6 API Middleware Display API ***AmpDisplay_DeleteWindow()***.

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 1-17. Returns for SDK6 Middleware Display API ***AmpDisplay_DeleteWindow()***.

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.5 AmpDisplay_GetDefaultCfg

API Syntax:

AmpDisplay_GetDefaultCfg (AMP_DISP_CFG_s * Cfg)

Function Description:

- This function is used to retrieve the default configuration in order to create a display handler.

Parameters:

Type	Parameter	Description
AMP_DISP_CFG_s *	**Cfg**	Display handler configuration parameters (Section 1.2.1.1)

Table 1-18. Parameters for SDK6 API Middleware Display API ***AmpDisplay_GetDefaultCfg()****.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 1-19. Returns for SDK6 Middleware Display API ***AmpDisplay_GetDefaultCfg()****.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.6 AmpDisplay_GetDefaultInitCfg

API Syntax:

AmpDisplay_GetDefaultInitCfg (AMP_DISP_INIT_CFG_s * Cfg)

Function Description:

- This function is used to retrieve the default initial configuration in order to initialize the display module.

Parameters:

Type	Parameter	Description
AMP_DISP_ INIT_CFG_s *	**Cfg**	Display module initialization configuration parameters (Section 1.2.6.1)

*Table 1-20. Parameters for SDK6 API Middleware Display API **AmpDisplay_GetDefaultInitCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-21. Returns for SDK6 Middleware Display API **AmpDisplay_GetDefaultInitCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.6.1 AmpDisplay_GetDefaultInitCfg > AMP_DISP_INIT_CFG_s

Type	Field	Description
UINT8 *	**MemoryPoolAddr**	Buffer start address used by the display module to create the display handler array
UINT32	**MemoryPoolSize**	Buffer size
UINT32	**MaxDeviceInterHdlr**	The desired number of device handlers
UINT32	**MaxWindowInterHdlr**	The total desired number of window handlers
UINT32	**MaxHdmiCheck**	The maximum number of HDMI check occurrences

*Table 1-22. Definition of **AMP_DISP_INIT_CFG_s** for Display API **AmpDisplay_GetDefaultInitCfg()**.*

1.2.7 AmpDisplay_GetDefaultWindowCfg

API Syntax:

AmpDisplay_GetDefaultWindowCfg (AMP_DISP_WINDOW_CFG_s * Cfg)

Function Description:

- This function is used to retrieve default window configuration parameters.

Parameters:

Type	Parameter	Description
AMP_DISP_ WINDOW_ CFG_s *	**Cfg**	Window configuration parameters (Section 1.2.2.1)

*Table 1-23. Parameters for SDK6 API Middleware Display API **AmpDisplay_GetDefaultWindowCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-24. Returns for SDK6 Middleware Display API **AmpDisplay_GetDefaultWindowCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.8 AmpDisplay_GetDeviceCfg

API Syntax:

AmpDisplay_GetDeviceCfg (AMP_DISP_HDLR_s * DispHdlr, AMP_DISP_DEV_CFG_s * DevCfg)

Function Description:

- This function is used to retrieve and send device information to a display handler.

Parameters:

Type	Parameter	Description
AMP_DISP_HDLR_s **	**DispHdlr**	The display handler instance for the user (Section 1.2.1.4)
AMP_DISP_DEV_CFG_s *	**DevCfg**	Device configuration parameters (Section 1.2.1.2)

*Table 1-25. Parameters for SDK6 API Middleware Display API **AmpDisplay_GetDeviceCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-26. Returns for SDK6 Middleware Display API **AmpDisplay_GetDeviceCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.9 AmpDisplay_GetInfo

API Syntax:

AmpDisplay_GetInfo (AMP_DISP_HDLR_s * DispHdlr, AMP_DISP_INFO_s * OutputInfo)

Function Description:

- This function is used to retrieve and send device/system information to the DSP.

Parameters:

Type	Parameter	Description
AMP_DISP_HDLR_s **	**DispHdlr**	The display handler instance for the user (Section 1.2.1.4)
AMP_DISP_INFO_s *	**OutputInfo**	Device VOUT information (Section 1.2.9.1)

*Table 1-27. Parameters for SDK6 API Middleware Display API **AmpDisplay_GetInfo()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-28. Returns for SDK6 Middleware Display API **AmpDisplay_GetInfo()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.9.1 AmpDisplay_GetInfo > AMP_DISP_INFO_s

Type	Field	Description
AMP_DISP_VOUT_INFO_s	**DeviceInfo**	Device VOUT information from the driver or DSP
AMP_DISP_WINDOW_HDLR_s *	**WindowHdlr**	Active window handler instance (Section 1.2.2.5)

*Table 1-29. Definition of **AMP_DISP_INFO_s** for Display API **AmpDisplay_GetInfo()**.*

1.2.10 AmpDisplay_GetWindowCfg

API Syntax:

AmpDisplay_GetWindowCfg (AMP_DISP_WINDOW_HDLR_s * WindowHdlr, AMP_DISP_WINDOW_CFG_s * Cfg)

Function Description:

- This function is used to retrieve window handler information.

Parameters:

Type	Parameter	Description
AMP_DISP_WINDOW_HDLR_s **	**WindowHdlr**	Window handler instance for the user (Section 1.2.2.5)
AMP_DISP_WINDOW_CFG_s *	**Cfg**	Window configuration parameters (Section 1.2.2.1)

*Table 1-30. Parameters for SDK6 API Middleware Display API **AmpDisplay_GetWindowCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-31. Returns for SDK6 Middleware Display API **AmpDisplay_GetWindowCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.11 AmpDisplay_Init

API Syntax:

AmpDisplay_Init (AMP_DISP_INIT_CFG_s * Cfg)

Function Description:

- This function is used to initialize the display module, as well as arrange display handler memory and initialize related flags.

Parameters:

Type	Parameter	Description
AMP_DISP_ INIT_CFG_s *	**Cfg**	Display module initialization configuration parameters (Section 1.2.6.1)

Table 1-32. Parameters for SDK6 API Middleware Display API ***AmpDisplay_Init()***.

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 1-33. Returns for SDK6 Middleware Display API ***AmpDisplay_Init()***.

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.12 AmpDisplay_SetDeviceCfg

API Syntax:

AmpDisplay_SetDeviceCfg (AMP_DISP_HDLR_s * DispHdlr, AMP_DISP_DEV_CFG_s * DevCfg)

Function Description:

- This function is used to set/update and send device information to a display handler. Takes effect upon starting the display.

Parameters:

Type	Parameter	Description
AMP_DISP_ HDLR_s **	**DispHdlr**	The display handler instance for the user (Section 1.2.1.3)
AMP_DISP_ DEV_CFG_s *	**DevCfg**	Device configuration parameters (Section 1.2.1.2)

*Table 1-34. Parameters for SDK6 API Middleware Display API **AmpDisplay_SetDeviceCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-35. Returns for SDK6 Middleware Display API **AmpDisplay_SetDeviceCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.13 AmpDisplay_SetWindowActivateFlag

API Syntax:

AmpDisplay_SetWindowActivateFlag (AMP_DISP_WINDOW_HDLR_s * WindowHdlr, UINT8 ActiveFlag)

Function Description:

- This function is used to activate a window on a device. Takes effect after **AmpDisplay_UpdateWindowState**.

Parameters:

Type	Parameter	Description
AMP_DISP_ WINDOW_ HDLR_s **	**WindowHdlr**	Window handler instance for the user to show OSD (Section 1.2.2.5)
UINT8	**ActiveFlag**	

*Table 1-36. Parameters for SDK6 API Middleware Display API **AmpDisplay_SetWindowActivateFlag()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-37. Returns for SDK6 Middleware Display API **AmpDisplay_SetWindowActivateFlag()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.14 AmpDisplay_SetWindowCfg

API Syntax:

AmpDisplay_SetWindowCfg (AMP_DISP_WINDOW_HDLR_s * WindowHdlr, AMP_DISP_WINDOW_CFG_s * Cfg)

Function Description:

- This function is used to set/update window handler information and send the event to the application to process if the relevant window is active.

Parameters:

Type	Parameter	Description
AMP_DISP_WINDOW_HDLR_s **	**WindowHdlr**	Window handler instance for the user (Section 1.2.2.5)
AMP_DISP_WINDOW_CFG_s *	**Cfg**	Updated window configuration parameters (Section 1.2.2.1)

*Table 1-38. Parameters for SDK6 API Middleware Display API **AmpDisplay_SetWindowCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-39. Returns for SDK6 Middleware Display API **AmpDisplay_SetWindowCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.15 AmpDisplay_Start

API Syntax:

AmpDisplay_Start (AMP_DISP_HDLR_s * DispHdlr)

Function Description:

- This function is used to set-up the device. The driver should be instructed to initialize the device, after which a callback event should be sent to the application.

Parameters:

Type	Parameter	Description
AMP_DISP_HDLR_s **	**DispHdlr**	The display handler instance for the user (Section 1.2.1.4)

*Table 1-40. Parameters for SDK6 API Middleware Display API **AmpDisplay_Start()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-41. Returns for SDK6 Middleware Display API **AmpDisplay_Start()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.16 AmpDisplay_Stop

API Syntax:

AmpDisplay_Stop (AMP_DISP_HDLR_s * DispHdlr)

Function Description:

- This function is used to disable the device. The driver should be instructed to shut-down the device, after which a callback event should be sent to the application.

Parameters:

Type	Parameter	Description
AMP_DISP_ HDLR_s **	**DispHdlr**	The display handler instance for the user (Section 1.2.1.4)

Table 1-42. Parameters for SDK6 API Middleware Display API ***AmpDisplay_Stop()***.

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 1-43. Returns for SDK6 Middleware Display API ***AmpDisplay_Stop()***.

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.17 AmpDisplay_Update

API Syntax:

AmpDisplay_Update (AMP_DISP_HDLR_s * DispHdlr)

Function Description:

- This function is used to update the window state on a display. Both **AmpDisplay_SetWindowCfg** and **AmpDisplay_ActivateWindow** take effect after this function is invoked.

Parameters:

Type	Parameter	Description
AMP_DISP_ HDLR_s **	**DispHdlr**	The display handler instance for the user (Section 1.2.1.4)

*Table 1-44. Parameters for SDK6 API Middleware Display API **AmpDisplay_Update()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-45. Returns for SDK6 Middleware Display API **AmpDisplay_Update()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

1.2.18 AmpDisplay_SetMaxVoutSize

API Syntax:

AmpDisplay_SetMaxVoutSize (UINT32 width, UINT32 height)

Function Description:

- This function is used to setup max vout0 size. The function MUST be invoked before middleware initialization if you want to enable seamless feature for memory allocation (seamless is an optional feature and may not be supported on every chip).

Parameters:

Type	Parameter	Description
UINT32	**Width**	Max width of out0
UINT32	**Height**	Max height of vout0

*Table 1-46. Parameters for SDK6 API Middleware Display API **AmpDisplay_SetMaxVoutSize()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 1-47. Returns for SDK6 Middleware Display API **AmpDisplay_SetMaxVout0Size()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2 OSD

2.1 OSD: Overview

This chapter provides information regarding on-screen display (OSD) APIs. The OSD command set provides functions used to set-up the on-screen display buffer, which should be prepared with data in the specified format. The DSP will blend the buffer according to video settings.

2.2 OSD: List of Functions

- (Section 2.2.1) AmpOsd_Create
- (Section 2.2.2) AmpOsd_Delete
- (Section 2.2.3) AmpOsd_GetBufferCfg
- (Section 2.2.4) AmpOsd_GetCfg
- (Section 2.2.5) AmpOsd_GetClutCfg
- (Section 2.2.6) AmpOsd_GetDefaultCfg
- (Section 2.2.7) AmpOsd_GetDefaultInitCfg
- (Section 2.2.8) AmpOsd_GetHdlrInfo
- (Section 2.2.9) AmpOsd_GetTransparentCfg
- (Section 2.2.10) AmpOsd_Init
- (Section 2.2.11) AmpOsd_SetBufferCfg
- (Section 2.2.12) AmpOsd_SetCfg
- (Section 2.2.13) AmpOsd_SetClutCfg
- (Section 2.2.14) AmpOsd_SetTransparentCfg

2.2.1 AmpOsd_Create

API Syntax:

AmpOsd_Create (AMP_OSD_CFG_s * cfg, AMP_OSD_HDLR_s ** osdHdlr)

Function Description:

- This function is used to create an OSD handler. Note that an OSD handler must be registered in order to manage OSD-related operations.

Parameters:

Type	Parameter	Description
AMP_OSD_ CFG_s *	**cfg**	OSD handler configuration parameters (Section 2.2.1.2)
AMP_OSD_ HDLR_s *	**osdHdlr**	OSD handler instance (Section 2.2.1.1)

*Table 2-1. Parameters for SDK6 API Middleware OSD API **AmpOsd_Create()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 2-2. Returns for SDK6 Middleware OSD API **AmpOsd_Create()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2.2.1.1 AmpOsd_ActivateOsdWindow > AMP_OSD_HDLR_s

Type	Field	Description
void *	**Ctx**	Pointer to codec context

*Table 2-3. Definition of **AMP_OSD_HDLR_s** for OSD API **AmpOsd_ActivateOsdWindow()**.*

2.2.1.2 AmpOsd_Create > AMP_OSD_CFG_s

Type	Field	Description
AMP_OSD_BUFFER_CFG_s	**BufCfg**	OSD buffer configuration (Section 2.2.1.3)
AMP_OSD_CLUT_CFG_s	**ColorLookupTable**	Color lookup table address (Section 2.2.1.4)
AMP_OSD_TRANSPARENT_CFG_s	**TransparentColor**	Transparency color configuration (Section 2.2.1.5)
AMP_DISP_OSD_HW_RESCALER_TYPE_e	**HwScalerType**	The hardware scaler type HW_OSD_RESCALER_NONE: No scale HW_OSD_RESCALER_INT: Scale to N times HW_OSD_RESCALER_ANY: Scale to any size (aligned) HW_OSD_RESCALER_UP: Only scale up
UINT8	**OsdBufRepeatField**	Repeat OSD buffer for top field and bottom field in interlaced mode
UINT8	**GlobalBlend**	Determines the OSD transparency color

*Table 2-4. Definition of **AMP_OSD_CFG_s** for OSD API **AmpOsd_Create()**.*

2.2.1.3 AmpOsd_Create > AMP_OSD_BUFFER_CFG_s

Type	Field	Description
AMP_DISP_OSD_FORMAT_e	**PixelFormat**	OSD buffer color format AMP_OSD_8BIT_CLUT_MODE: 8-bit color look-up table mode AMP_OSD_16BIT_VYU_RGB_565: VYU_RGB_565 AMP_OSD_16BIT_UYV_BGR_565: UYV_BGR_565 AMP_OSD_16BIT_AYUV_4444: AYUV_4444 AMP_OSD_16BIT_RGBA_4444: RGBA_4444 AMP_OSD_16BIT_BGRA_4444: BGRA_4444 AMP_OSD_16BIT_ABGR_4444: ABGR_4444 AMP_OSD_16BIT_ARGB_4444: ARGB_4444 AMP_OSD_16BIT_AYUV_1555: AYUV_1555 AMP_OSD_16BIT_YUV_1555: YUV_1555, MSB ignored AMP_OSD_16BIT_RGBA_5551: RGBA_5551 AMP_OSD_16BIT_BGRA_5551: BGRA_5551 AMP_OSD_16BIT_ABGR_1555: ABGR_1555 AMP_OSD_16BIT_ARGB_1555: ARGB_1555 AMP_OSD_32BIT_AYUV_8888: AYUV_8888 AMP_OSD_32BIT_RGBA_8888: RGBA_8888 AMP_OSD_32BIT_BGRA_8888: BGRA_8888 AMP_OSD_32BIT_ABGR_8888: ABGR_8888 AMP_OSD_32BIT_ARGB_8888: ARGB_8888
UINT32	**BufWidth**	OSD buffer width
UINT32	**BufHeight**	OSD buffer height
UINT32	**BufPitch**	OSD buffer pitch
UINT32	**BufAddr**	Current buffer

*Table 2-5. Definition of **AMP_OSD_BUFFER_CFG_s** for OSD API **AmpOsd_Create()**.*

2.2.1.4 AmpOsd_Create > AMP_OSD_CLUT_CFG_s

Type	Field	Description
UINT8 *	**RgbTable**	CLUT RGB Table
UINT8 *	**BlendTable**	CLUT Blend Table
UINT8 *	**YuvTable**	CLUT YUV Table

*Table 2-6. Definition of **AMP_OSD_CLUT_CFG_s** for OSD API **AmpOsd_Create()**.*

2.2.1.5 AmpOsd_Create > AMP_OSD_TRANSPARENT_CFG_s

Type	Field	Description
UINT8	**Enable**	Enable transparency
UINT16	**Color**	Transparency color (one only)

*Table 2-7. Definition of **AMP_OSD_TRANSPARENT_CFG_s** for OSD API **AmpOsd_Create()**.*

2.2.2 AmpOsd_Delete

API Syntax:

AmpOsd_Delete (AMP_OSD_HDLR_s * osdHdlr)

Function Description:

- This function is used to delete an OSD handler, which clears the OSD handler information and de-registers the handler.

Parameters:

Type	Parameter	Description
AMP_OSD_HDLR_s *	**osdHdlr**	OSD handler instance (Section 2.2.1.1)

*Table 2-8. Parameters for SDK6 API Middleware OSD API **AmpOsd_Delete()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 2-9. Returns for SDK6 Middleware OSD API **AmpOsd_Delete()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2.2.3 AmpOsd_GetBufferCfg

API Syntax:

AmpOsd_GetBufferCfg (AMP_OSD_HDLR_s * osdHdlr, AMP_OSD_BUFFER_CFG_s * cfg)

Function Description:

- This function is used to retrieve the buffer-related configuration settings of a specified OSD handler.

Parameters:

Type	Parameter	Description
AMP_OSD_HDLR_s *	**osdHdlr**	OSD handler instance (Section 2.2.1.1)
AMP_OSD_BUFFER_CFG_s *	**cfg**	OSD configuration parameters (Section 2.2.1.3)

*Table 2-10. Parameters for SDK6 API Middleware OSD API **AmpOsd_GetBufferCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 2-11. Returns for SDK6 Middleware OSD API **AmpOsd_GetBufferCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2.2.4 AmpOsd_GetCfg

API Syntax:

AmpOsd_GetCfg (AMP_OSD_HDLR_s * osdHdlr, AMP_OSD_CFG_s * cfg)

Function Description:

- This function is used to retrieve OSD configuration settings of a specified OSD handler.

Parameters:

Type	Parameter	Description
AMP_OSD_HDLR_s *	**osdHdlr**	OSD handler instance (Section 2.2.1.1)
AMP_OSD_CFG_s *	**cfg**	OSD configuration parameters (Section 2.2.1.2)

*Table 2-12. Parameters for SDK6 API Middleware OSD API **AmpOsd_GetCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 2-13. Returns for SDK6 Middleware OSD API **AmpOsd_GetCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2.2.5 AmpOsd_GetClutCfg

API Syntax:

AmpOsd_GetClutCfg (AMP_OSD_HDLR_s * osdHdlr, AMP_OSD_CLUT_CFG_s * cfg)

Function Description:

- This function is used to retrieve the CLUT configuration settings of a specified OSD handler.

Parameters:

Type	Parameter	Description
AMP_OSD_ HDLR_s *	**osdHdlr**	OSD handler instance (Section 2.2.1.1)
AMP_OSD_ CLUT_CFG_s *	**cfg**	OSD CLUT configuration parameters (Section 2.2.1.4)

*Table 2-14. Parameters for SDK6 API Middleware OSD API **AmpOsd_GetClutCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 2-15. Returns for SDK6 Middleware OSD API **AmpOsd_GetClutCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2.2.6 AmpOsd_GetDefaultCfg

API Syntax:

AmpOsd_GetDefaultCfg (AMP_OSD_CFG_s * cfg)

Function Description:

- This function is used to retrieve default configuration settings for the creation of an OSD handler.

Parameters:

Type	Parameter	Description
AMP_OSD_ CFG_s *	**cfg**	OSD handler configuration parameters (Section 2.2.1.2)

*Table 2-16. Parameters for SDK6 API Middleware OSD API **AmpOsd_GetDefaultCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 2-17. Returns for SDK6 Middleware OSD API **AmpOsd_GetDefaultCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2.2.7 AmpOsd_GetDefaultInitCfg

API Syntax:

AmpOsd_GetDefaultInitCfg (AMP_OSD_INIT_CFG_s * cfg)

Function Description:

- This function is used to retrieve default initial configuration settings for the initialization of the OSD mini-module.

Parameters:

Type	Parameter	Description
AMP_OSD_CFG_s *	**cfg**	Initialization configuration parameters (Section 2.2.7.1)

*Table 2-18. Parameters for SDK6 API Middleware OSD API **AmpOsd_GetDefaultInitCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 2-19. Returns for SDK6 Middleware OSD API **AmpOsd_GetDefaultInitCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2.2.7.1 AmpOsd_GetDefaultInitCfg > AMP_OSD_INIT_CFG_s

Type	Field	Description
UINT8 *	**MemoryPoolAddr**	Buffer start address used by the OSD module to create the OSD handler array
UINT32	**MemoryPoolSize**	Size of the buffer
UINT32	**MaxOsdInterHdlr**	The desired number of OSD handlers

*Table 2-20. Definition of **AMP_OSD_INIT_CFG_s** for OSD API **AmpOsd_GetDefaultInitCfg()**.*

2.2.8 AmpOsd_GetHdlrInfo

API Syntax:

AmpOsd_GetHdlrInfo (int channel, AMP_OSD_HDLR_INFO_s * outputInfo)

Function Description:

- This function is used to retrieve the OSD active handler system information.

Parameters:

Type	Parameter	Description
int	**channel**	The channel on which the OSD is shown
AMP_OSD_ HDLR_INFO_s *	**OutputInfo**	Returns OSD active handler and window handler information (Section 2.2.8.1)

*Table 2-21. Parameters for SDK6 API Middleware OSD API **AmpOsd_GetHdlrInfo()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 2-22. Returns for SDK6 Middleware OSD API **AmpOsd_GetHdlrInfo()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2.2.8.1 AmpOsd_GetHdlrInfo > AMP_OSD_HDLR_INFO_s

Type	Field	Description
AMP_OSD_HDLR_s *	**OsdHdlr**	Active OSD handler instance (Section 2.2.1.1)
AMP_DISP_WINDOW_HDLR_s *	**WindowHdlr**	Active window handler instance (Section 2.2.1.2)

*Table 2-23. Definition of **AMP_OSD_HDLR_INFO_s** for OSD API **AmpOsd_GetHdlrInfo()**.*

2.2.9 AmpOsd_GetTransparentCfg

API Syntax:

AmpOsd_GetTransparentCfg (AMP_OSD_HDLR_s * osdHdlr, AMP_OSD_TRANSPARENT_CFG_s * cfg)

Function Description:

- This function is used to retrieve transparency configuration settings for the OSD handler.

Parameters:

Type	Parameter	Description
AMP_OSD_HDLR_s *	**osdHdlr**	OSD handler instance (Section 2.2.1.1)
AMP_OSD_TRANSPARENT_CFG_s *	**cfg**	OSD transparency configuration parameters (Section 2.2.1.5)

*Table 2-24. Parameters for SDK6 API Middleware OSD API **AmpOsd_GetTransparentCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 2-25. Returns for SDK6 Middleware OSD API **AmpOsd_GetTransparentCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2.2.10 AmpOsd_Init

API Syntax:

AmpOsd_Init (AMP_OSD_INIT_CFG_s * cfg)

Function Description:

- This function is used to initialize the OSD mini-module, arrange OSD handler memory, and initialize related flags.

Parameters:

Type	Parameter	Description
AMP_OSD_ CFG_s *	**cfg**	Initialization configuration parameters (Section 2.2.7.1)

*Table 2-26. Parameters for SDK6 API Middleware OSD API **AmpOsd_Init()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 2-27. Returns for SDK6 Middleware OSD API **AmpOsd_Init()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2.2.11 AmpOsd_SetBufferCfg

API Syntax:

AmpOsd_SetBufferCfg (AMP_OSD_HDLR_s * osdHdlr, AMP_OSD_BUFFER_CFG_s * cfg)

Function Description:

- This function is used to set/update the buffer-related configuration settings of a specified OSD handler. When the OSD buffer configuration information is updated in the OSD handler, the new settings will take effect immediately (assuming OSD is active).

Parameters:

Type	Parameter	Description
AMP_OSD_HDLR_s *	**osdHdlr**	OSD handler instance (Section 2.2.1.1)
AMP_OSD_BUFFER_CFG_s *	**cfg**	OSD configuration parameters (Section 2.2.1.3)

*Table 2-28. Parameters for SDK6 API Middleware OSD API **AmpOsd_SetBufferCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 2-29. Returns for SDK6 Middleware OSD API **AmpOsd_SetBufferCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2.2.12 AmpOsd_SetCfg

API Syntax:

AmpOsd_SetCfg (AMP_OSD_HDLR_s * osdHdlr, AMP_OSD_CFG_s * cfg)

Function Description:

- This function is used to set/update OSD configuration settings of a specified OSD handler. When the OSD configuration information is updated in the OSD handler, the new settings will take effect immediately (assuming OSD is active).
-

Parameters:

Type	Parameter	Description
AMP_OSD_HDLR_s *	**osdHdlr**	OSD handler instance (Section 2.2.1.1)
AMP_OSD_CFG_s *	**cfg**	OSD configuration parameters (Section 2.2.1.2)

*Table 2-30. Parameters for SDK6 API Middleware OSD API **AmpOsd_SetCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 2-31. Returns for SDK6 Middleware OSD API **AmpOsd_SetCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2.2.13 AmpOsd_SetClutCfg

API Syntax:

AmpOsd_SetClutCfg (AMP_OSD_HDLR_s * osdHdlr, AMP_OSD_CLUT_CFG_s * cfg)

Function Description:

- This function is used to set/update the CLUT configuration settings of a specified OSD handler. When the CLUT configuration information is updated in the OSD handler, the new settings will take effect immediately (assuming OSD is active).

Parameters:

Type	Parameter	Description
AMP_OSD_ HDLR_s *	**osdHdlr**	OSD handler instance (Section 2.2.1.1)
AMP_OSD_ CLUT_CFG_s *	**cfg**	OSD CLUT configuration parameters (Section 2.2.1.4)

Table 2-32. Parameters for SDK6 API Middleware OSD API ***AmpOsd_SetClutCfg()****.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 2-33. Returns for SDK6 Middleware OSD API ***AmpOsd_SetClutCfg()****.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

2.2.14 AmpOsd_SetTransparentCfg

API Syntax:

AmpOsd_SetTransparentCfg (AMP_OSD_HDLR_s * osdHdlr, AMP_OSD_TRANSPARENT_CFG_s * cfg)

Function Description:

- This function is used to set/update transparency configuration settings for the OSD handler. When the transparency configuration information is updated in the OSD handler, the new settings will take effect immediately (assuming OSD is active).

Parameters:

Type	Parameter	Description
AMP_OSD_ HDLR_s *	**osdHdlr**	OSD handler instance (Section 2.2.1.1)
AMP_OSD_ TRANSPAR-ENT_CFG_s *	**cfg**	OSD transparency configuration parameters (Section 2.2.1.5)

*Table 2-34. Parameters for SDK6 API Middleware OSD API **AmpOsd_SetTransparentCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 2-35. Returns for SDK6 Middleware OSD API **AmpOsd_SetTransparentCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

3 FIFO

3.1 FIFO: Overview

This chapter provides information regarding the bitstream descriptor manager.

Features include:

- FIFO is used to manage the bits buffer descriptor
- FIFO enables handshaking between the format and the codec
- More than one virtual FIFO can be created to read data from the base FIFO

3.2 FIFO: List of Functions

- (Section 3.2.1) AmpFifo_Create
- (Section 3.2.2) AmpFifo_Delete
- (Section 3.2.3) AmpFifo_EraseAll
- (Section 3.2.4) AmpFifo_GetDefaultCfg
- (Section 3.2.5) AmpFifo_GetInfo
- (Section 3.2.6) AmpFifo_GetInitDefaultCfg
- (Section 3.2.7) AmpFifo_Init
- (Section 3.2.8) AmpFifo_PeekEntry
- (Section 3.2.9) AmpFifo_PrepareEntry
- (Section 3.2.10) AmpFifo_RemoveEntry
- (Section 3.2.11) AmpFifo_WriteEntry
- (Section 3.2.12) AmpFifo_Resume

3.2.1 AmpFifo_Create

API Syntax:

AmpFifo_Create (AMP_FIFO_CFG_s * cfg, AMP_FIFO_HDLR_s ** fifo)

Function Description:

- This function is used to create a virtual FIFO on a codec, manage the bits descriptor generated by the codec, or feed data to the codec. On creation, the **getNumDesc()** function of the codec will be invoked to allocate the virtual descriptor queue in the FIFO.
- Note that only one write FIFO per codec is permitted.

Parameters:

Type	Parameter	Description
AMP_FIFO_ CFG_s *	**cfg**	FIFO configuration parameters (Section 3.2.1.1)
AMP_FIFO_ HDLR_s **	**fifo**	The FIFO handler instance (Section 3.2.1.3)

*Table 3-1. Parameters for SDK6 API Middleware FIFO API **AmpFifo_Create()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 3-2. Returns for SDK6 Middleware FIFO API **AmpFifo_Create()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

3.2.1.1 AmpFifo_Create > AMP_FIFO_CFG_s

Type	Field	Description
void *	**hCodec**	The codec the FIFO is working on
UINT32	**NumEntries**	Number of entries of a FIFO
UINT32	**IsVirtual**	Creating virtual FIFO (for data flow) or not (for codec)
AMP_CALLBACK_f	**cbEvent**	The callback function for FIFO event
UINT32	**EventDataConsumedThreshold**	The threshold for event **AMP_FIFO_CALLBACK_EVENT_DATA_CONSUMED**. The event is only triggered if the remaining data is under the threshold. 0 to disable.
UINT32	**EventDataReadySkipNum**	If not 0, **AMP_FIFO_CALLBACK_EVENT_DATA_READY** will not be triggered for every frame encoded. Instead, it will be triggered every (**eventDataReadySkipNum** frame+1) frames.
AMP_CALLBACK_f	**cbGetWritePoint**	The callback function invoked when **prepareSpace** is received. Note that this should only be used on the FIFO linked to the codec. Only takes effect on a codec read FIFO. The callback should be registered by a decode manager for all decode codecs.
UINT8	**SyncRpOnWrite**	If 0, RP of base FIFO will be updated on removal of the virtual FIFO. If 1, RP of base FIFO will be updated on write of base FIFO. ONLY takes effect on base FIFO.
UINT32	**RawBaseAddr**	When sync RP on write is on and RawBaseAddr/RawLimitAddr is given, base fifo will remove desc automatically on overwrite. Only needed on base FIFO and sync on write is on.
UINT32	**RawLimitAddr**	When sync RP on write is on and RawBaseAddr/RawLimitAddr is given, base FIFO will remove desc automatically on overwrite. Only needed on base FIFO and sync on write is on.
UINT64	**TickPerSecond**	TickPerSecond used for PTS for the FIFO. Could be 0 if we would NOT like to use init data.
AMP_FIFO_CFG_INIT_DATA_s	**InitData**	Init status for virtual FIFO

*Table 3-3. Definition of **AMP_FIFO_CFG_s** for FIFO API **AmpFifo_Create()**.*

3.2.1.2 AmpFifo_Create > AMP_FIFO_CFG_INIT_DATA_s

Type	Field	Description
UINT8	**CreateFifoWithInitData**	If create FIFO with init data (if data valid)
AMP_FIFO_CFG_INIT_DATA_FETCH_CONDITION_e	**InitCondition**	Defines how to descript initial data required.
UINT64	**TimeLength**	In normal case, the value of backward fetch is 0. This means FIFO will output frames only after FIFO is created. As the value is not 0, FIFO will try to determine the longest valid frames
UINT64	**NumFrame**	Number of frame
UINT64	**StartTime**	Start time in ms
AMP_FIFO_FRMAE_TYPE_e	**FirstFrameType**	Frame type of first frame for valid data; used on video stream to ensure start with IDR.

Type	Field	Description
UINT64	**OnCreateFirstFramePts**	[OUT] PTS of first frame; 0xFFFFFFFFFFFFFFFF if first frame is not valid on create
UINT64	**OnCreateTimeLength**	[OUT] Data length in FIFO on create in ms

*Table 3-4. Definition of **AMP_FIFO_CFG_INIT_DATA_s** for FIFO API **AmpFifo_Create()**.*

3.2.1.3 AmpFifo_Create > AMP_FIFO_s_AMP_FIFO_HDLR_s

Type	Field	Description
UINT32	**nFifoId**	Unique FIFO ID
void *	**Ctx**	Private data of the FIFO

*Table 3-5. Definition of **AMP_FIFO_s_AMP_FIFO_HDLR_s** for FIFO API **AmpFifo_Create()**.*

3.2.2 AmpFifo_Delete

API Syntax:

AmpFifo_Delete (AMP_FIFO_HDLR_s * fifo)

Function Description:

- This function is used to close a FIFO.

Parameters:

Type	Parameter	Description
AMP_FIFO_ HDLR_s **	**fifo**	The FIFO to be closed (Section 3.2.1.3)

*Table 3-6. Parameters for SDK6 API Middleware FIFO API **AmpFifo_Delete()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 3-7. Returns for SDK6 Middleware FIFO API **AmpFifo_Delete()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

3.2.3 AmpFifo_EraseAll

API Syntax:

AmpFifo_EraseAll (AMP_FIFO_HDLR_s * fifo)

Function Description:

- This function is used to erase all data in a specified FIFO. Resets the read and write points.

Parameters:

Type	Parameter	Description
AMP_FIFO_ HDLR_s **	**fifo**	The target FIFO (Section 3.2.1.3)

Table 3-8. Parameters for SDK6 API Middleware FIFO API ***AmpFifo_EraseAll()***.

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 3-9. Returns for SDK6 Middleware FIFO API ***AmpFifo_EraseAll()***.

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

3.2.4 AmpFifo_GetDefaultCfg

API Syntax:

AmpFifo_GetDefaultCfg (AMP_FIFO_CFG_s * defaultCfg)

Function Description:

- This function is used to retrieve the default FIFO configuration settings.

Parameters:

Type	Parameter	Description
AMP_FIFO_ CFG_s *	**defaultCfg**	FIFO configuration parameters (Section 3.2.1.1)

Table 3-10. Parameters for SDK6 API Middleware FIFO API ***AmpFifo_GetDefaultCfg()***.

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 3-11. Returns for SDK6 Middleware FIFO API ***AmpFifo_GetDefaultCfg()***.

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

3.2.5 AmpFifo_GetInfo

API Syntax:

AmpFifo_GetInfo (AMP_FIFO_HDLR_s * fifo, AMP_FIFO_INFO_s * info)

Function Description:

- This function is used to retrieve FIFO-related information.

Parameters:

Type	Parameter	Description
AMP_FIFO_HDLR_s **	**fifo**	The target FIFO (Section 3.2.1.3)
AMP_FIFO_INFO_s *	**info**	FIFO information (Section 3.2.5.1)

*Table 3-12. Parameters for SDK6 API Middleware FIFO API **AmpFifo_GetInfo()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 3-13. Returns for SDK6 Middleware FIFO API **AmpFifo_GetInfo()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

3.2.5.1 AmpFifo_GetInfo > AMP_FIFO_INFO_s

Type	Field	Description
UINT32	**TotalEntries**	Total number of entries
UINT32	**AvailEntries**	Number of entries containing data

*Table 3-14. Definition of **AMP_FIFO_INFO_s** for FIFO API **AmpFifo_GetInfo()**.*

3.2.6 AmpFifo_GetInitDefaultCfg

API Syntax:

AmpFifo_GetInitDefaultCfg (AMP_FIFO_INIT_CFG_s * defaultCfg)

Function Description:

- This function is used to retrieve the default FIFO configuration settings for initialization.

Parameters:

Type	Parameter	Description
AMP_FIFO_ INIT_CFG_s *	**defaultCfg**	FIFO configuration parameters (Section 3.2.6.1)

*Table 3-15. Parameters for SDK6 API Middleware FIFO API **AmpFifo_GetInitDefaultCfg()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 3-16. Returns for SDK6 Middleware FIFO API **AmpFifo_GetInitDefaultCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

3.2.6.1 AmpFifo_GetInitDefaultCfg > AMP_FIFO_INIT_CFG_s

Type	Field	Description
UINT8 *	**MemoryPoolAddr**	Buffer start address for FIFO module including descriptor
UINT32	**MemoryPoolSize**	Buffer size
UINT32	**NumMaxFifo**	Maximum supported number of virtual FIFOs
UINT32	**NumMaxCodec**	Maximum supported number of codecs
AMP_TASK_INFO_s	**TaskInfo**	Task-related information (Section 3.2.6.2)

*Table 3-17. Definition of **AMP_FIFO_INIT_CFG_s** for FIFO API **AmpFifo_GetInitDefaultCfg()**.*

3.2.6.2 AmpFifo_GetInitDefaultCfg > AMP_TASK_INFO_s

Type	Field	Description
UINT32	**Priority**	Task priority
UINT32	**StackSize**	Task stack size

*Table 3-18. Definition of **AMP_TASK_INFO_s** for FIFO API **AmpFifo_GetInitDefaultCfg()**.*

3.2.7 AmpFifo_Init

API Syntax:

AmpFifo_Init (AMP_FIFO_INIT_CFG_s * cfg)

Function Description:

- This function is used to initialize the FIFO module. Note that this function should only be invoked once, and must be invoked prior to using the FIFO module. The memory pool for the module will be provided by the user.

Parameters:

Type	Parameter	Description
AMP_FIFO_ INIT_CFG_s *	**cfg**	FIFO configuration parameters (Section 3.2.6.1)

Table 3-19. Parameters for SDK6 API Middleware FIFO API ***AmpFifo_Init()***.

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 3-20. Returns for SDK6 Middleware FIFO API ***AmpFifo_Init()***.

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

3.2.8 AmpFifo_PeekEntry

API Syntax:

AmpFifo_PeekEntry (AMP_FIFO_HDLR_s * fifo, AMP_BITS_DESC_s ** desc, UINT32 distanceToLastEntry)

Function Description:

- This function is used to peek at a target FIFO entry using a specified distance.
- For example, if there are three entries in the current FIFO:
  - (wp) [entry A] [entry B] [entry C] (rp)
  - **AmpFifo_PeekEntry**(hFifo, &desc, 0) will retrieve entry C ;
  - **AmpFifo_PeekEntry**(hFifo, &desc, 1) will retrieve entry B ;
  - **AmpFifo_PeekEntry**(hFifo, &desc, 2) will retrieve entry A ;
  - **AmpFifo_PeekEntry**(hFifo, &desc, 3) will return as failed.

Parameters:

Type	Parameter	Description
AMP_FIFO_HDLR_s *	**fifo**	The target FIFO (Section 3.2.1.3)
AMP_BITS_DESC_s **	**desc**	The specified entry (Section 3.2.8.1)
UINT32	**distanceToLastEntry**	Distance to the last entry

*Table 3-21. Parameters for SDK6 API Middleware FIFO API **AmpFifo_PeekEntry()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 3-22. Returns for SDK6 Middleware FIFO API **AmpFifo_PeekEntry()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

3.2.8.1 AmpFifo_PeekEntry > AMP_BITS_DESC_s

Type	Field	Description
UINT32	**SeqNum**	Sequential number of bits buffer
UINT32	**Pts**	Presentation time stamp in ticks
AMP_FIFO_FRAME_TYPE_e	**Type**	Entry data type AMP_FIFO_TYPE_IDR_FRAME: IDR frame type. AMP_FIFO_TYPE_I_FRAME: I frame type AMP_FIFO_TYPE_P_FRAME: P frame type AMP_FIFO_TYPE_B_FRAME: B frame type AMP_FIFO_TYPE_JPEG_FRAME: JPEG main frame AMP_FIFO_TYPE_THUMBNAIL_FRAME: JPEG screennail frame AMP_FIFO_TYPE_SCREENNAIL_FRAME: JPEG thumbnail frame AMP_FIFO_TYPE_AUDIO_FRAME: Audio frame AMP_FIFO_TYPE_UNDEFINED: Others AMP_FIFO_TYPE_EOS_MARK: Special frame type to indicate state of feeding. AMP_FIFO_TYPE_DECODE_MARK: Used when feeding bit-stream to DSP will push out all frames. AMP_FIFO_TYPE_EOS: EOS bits that feed to raw buffer
UINT8	**Completed**	Whether the buffer contains a complete entry
UINT16	**Align**	
UINT8 *	**StartAddr**	Data size alignment (in bytes, align = 2^n, where n is a integer) start address of data
UINT32	**Size**	Actual data size

*Table 3-23. Definition of **AMP_BITS_DESC_s** for FIFO API **AmpFifo_PeekEntry()**.*

3.2.9 AmpFifo_PrepareEntry

API Syntax:

AmpFifo_PrepareEntry (AMP_FIFO_HDLR_s * fifo, AMP_BITS_DESC_s * desc)

Function Description:

- This function is used to prepare space to write data to FIFO. Note that this function must be used to retrieve the correct descriptor of the write point.

Parameters:

Type	Parameter	Description
AMP_FIFO_HDLR_s *	**fifo**	The target FIFO (Section 3.2.1.3)
AMP_BITS_DESC_s *	**desc**	The descriptor of the space (Section 3.2.8.1)

*Table 3-24. Parameters for SDK6 API Middleware FIFO API **AmpFifo_PrepareEntry()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 3-25. Returns for SDK6 Middleware FIFO API **AmpFifo_PrepareEntry()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

3.2.10 AmpFifo_RemoveEntry

API Syntax:

AmpFifo_RemoveEntry (AMP_FIFO_HDLR_s * fifo, UINT32 n)

Function Description:

- This function is used to remove the last **n** entry.
- For example, if there are three entries in the current FIFO:
  (wp) [entry A] [entry B] [entry C] (rp)
  After **AmpFifo_RemoveEntry**(hFifo, 2)
  it becomes
  (wp) [entry A] (rp)
- Note that the remove function only operates on the virtual FIFO indicated by the **n** parameter. If there are other FIFOs operating on the same codec, their read points will not be changed.

Parameters:

Type	Parameter	Description
AMP_FIFO_ HDLR_s *	**fifo**	The target FIFO (Section 3.2.1.3)
UINT32	**n**	The number of the entry to be removed

*Table 3-26. Parameters for SDK6 API Middleware FIFO API **AmpFifo_RemoveEntry()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 3-27. Returns for SDK6 Middleware FIFO API **AmpFifo_RemoveEntry()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

3.2.11 AmpFifo_WriteEntry

API Syntax:

AmpFifo_WriteEntry (AMP_FIFO_HDLR_s * fifo, AMP_BITS_DESC_s * desc)

Function Description:

- This function is used to write data to a FIFO.
- For example, if there is one entry in the current FIFO:

  (wp) [entry A] (rp)
  After AmpFifo_WriteDataToFifo(hFifo, descEntryB)
  it becomes
  (wp) [entry B][entry A] (rp)
- The write points of all FIFOs working on the same codec will be updated after this function is invoked.

Parameters:

Type	Parameter	Description
AMP_FIFO_ HDLR_s *	**fifo**	The target FIFO (Section 3.2.1.3)
AMP_BITS_ DESC_s *	**desc**	The descriptor of the data to write (Section 3.2.8.1)

*Table 3-28. Parameters for SDK6 API Middleware FIFO API **AmpFifo_WriteEntry()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 3-29. Returns for SDK6 Middleware FIFO API **AmpFifo_WriteEntry()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

3.2.12 AmpFifo_Resume

API Syntax:

AmpFifo_Resume (AMP_FIFO_HDLR_s * fifo, AMP_FIFO_CFG_INIT_DATA_s * initData)

Function Description:

- This function is used to resume a FIFO created with the init condition paused.
- We could apply initial condition on resume as start time/number of frames

Parameters:

Type	Parameter	Description
AMP_FIFO_ HDLR_s *	**fifo**	The target FIFO (Section 3.2.1.3)
AMP_FIFO_ CFG_INIT_ DATA_s *	**initData**	Init data on resume (Section 3.2.1.2)

*Table 3-30. Parameters for SDK6 API Middleware FIFO API **AmpFifo_Resume()**.*

Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 3-31. Returns for SDK6 Middleware FIFO API **AmpFifo_Resume()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4 ImgDzoom

4.1 ImgDzoom: Overview

This chapter provides the APIs for Digital Zoom control module implementation, including:

1. Init Digital Zoom control module function
2. Create Digital Zoom control module function
3. Delete Digital Zoom control module function
4. Reset Digital Zoom control module current context
5. Callback of VIN ISR handler
6. Digital Zoom control APIs
7. Get Digital Zoom module information

4.2 ImgDzoom: List of APIs

- AmpImgDzoom_ChangeResolutionHandler
- AmpImgDzoom_Create
- AmpImgDzoom_Delete
- AmpImgDzoom_GetDefaultCfg
- AmpImgDzoom_GetDzoomInfo
- AmpImgDzoom_GetDzoomStatus
- AmpImgDzoom_GetInitDefaultCfg
- AmpImgDzoom_Init
- AmpImgDzoom_PresetDzoomJump
- AmpImgDzoom_RegDzoomTable
- AmpImgDzoom_ResetStatus
- AmpImgDzoom_SetDzoomJump
- AmpImgDzoom_SetDzoomPosition
- AmpImgDzoom_StopDzoom
- AmpImgDzoom_Enable
- AmpImgDzoom_MainTask

4.2.1 AmpImgDzoom_ChangeResolutionHandler

API Syntax:

AmpImgDzoom_ChangeResolutionHandler (AMP_IMG_DZOOM_HDLR_s * ImgDzoom, AMP_IMG_DZOOM_VIN_INVALID_INFO_s * Info)

Function Description:

- This function is used to change resolution for ImgDzoom.

Parameters:

Type	Parameter	Description
AMP_IMG_DZOOM_HDLR_s *	**ImgDzoom**	ImgDzoom. (**AMP_IMG_DZOOM_HDLR_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.1.1 for more details.
AMP_IMG_DZOOM_VIN_INVALID_INFO_s *	**Info**	VinInvalidInfo. (**AMP_IMG_DZOOM_VIN_INVALID_INFO_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.1.2 for more details.

*Table 4-1. Parameters for ImgDzoom SDK6 API **AmpImgDzoom_ChangeResolutionHandler()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-2. Returns for ImgDzoom SDK6 API **AmpImgDzoom_ChangeResolutionHandler()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4.2.1.1 AmpImgDzoom_ChangeResolutionHandler > AMP_IMG_DZOOM_HDLR_s

Type	Field	Description
void *	**Ctx**	Private data of the dzoom

*Table 4-3. Definition of **AMP_IMG_DZOOM_HDLR_s** for ImgDzoom SDK6 API **AmpImgDzoom_ChangeResolutionHandler()**.*

4.2.1.2 AmpImgDzoom_ChangeResolutionHandler > AMP_IMG_DZOOM_VIN_INVALID_INFO_s

Type	Field	Description
UINT32	**CapW**	Capture Width
UINT32	**CapH**	Capture Height

*Table 4-4. Definition of **AMP_IMG_DZOOM_VIN_INVALID_INFO_s** for ImgDzoom SDK6 API **AmpImgDzoom_ChangeResolutionHandler()**.*

4.2.2 AmpImgDzoom_Create

API Syntax:

AmpImgDzoom_Create (AMP_IMG_DZOOM_CFG_s *Cfg, AMP_IMG_DZOOM_HDLR_s * * ImgDzoom)

Function Description:

- This function is used for the ImgDzoom handler creation.

Parameters:

Type	Parameter	Description
AMP_IMG_ DZOOM_CFG_s*	**Cfg**	ImgDzoom module config. (**AMP_IMG_DZOOM_CFG_s** is defined in `ImgDz oom.h`) Please refer to Section 4.2.2.1 for more details.
AMP_IMG_ DZOOM_ HDLR_s * *	**ImgDzoom**	ImgDzoom to create. (**AMP_IMG_DZOOM_HDLR_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.1.1 for more details.

*Table 4-5. Parameters for ImgDzoom SDK6 API **AmpImgDzoom_Create()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-6. Returns for ImgDzoom SDK6 API **AmpImgDzoom_Create()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes and **AMP_BITS_DESC_s**.

4.2.2.1 AmpImgDzoom_Create > AMP_IMG_DZOOM_CFG_s

Type	Type	Description
AMP_TASK_ INFO_s	**TaskInfo**	Task information
UINT32	**MaxDzoomFactor**	The maximum dzoom factor
UINT32	**ImgModeContextId**	The image mode context Id
UINT32	**ImgModeBatchId**	The image mode batch Id

*Table 4-7. Definition of **AMP_IMG_DZOOM_CFG_s** for ImgDzoom SDK6 API **AmpImgDzoom_Create()**.*

4.2.3 AmpImgDzoom_Delete

API Syntax:

AmpImgDzoom_Delete (AMP_IMG_DZOOM_HDLR_s * ImgDzoom)

Function Description:

- This function is used to delete the image dzoom task of this handler.

Parameters:

Type	Parameter	Description
AMP_IMG_ DZOOM_ HDLR_s *	**ImgDzoom**	ImgDzoom to delete. (**AMP_IMG_DZOOM_HDLR_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.1.1 for more details.

*Table 4-8. Parameters for ImgDzoom SDK6 API **AmpImgDzoom_Delete()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-9. Returns for ImgDzoom SDK6 API **AmpImgDzoom_Delete()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4.2.4 AmpImgDzoom_GetDefaultCfg

API Syntax:

AmpImgDzoom_GetDefaultCfg (AMP_IMG_DZOOM_CFG_s * DefaultCfg)

Function Description:

- This function is used to get the ImgDzoom handler default configuration.

Parameters:

Type	Parameter	Description
AMP_IMG_ DZOOM_CFG_s *	**DefaultCfg**	ImgDzoom module config. (**AMP_IMG_DZOOM_CFG_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.2.1 for more details.

*Table 4-10. Parameters for ImgDzoom SDK6 API **AmpImgDzoom_GetDefaultCfg()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-11. Returns for ImgDzoom SDK6 API **AmpImgDzoom_GetDefaultCfg()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4.2.5 AmpImgDzoom_GetDzoomInfo

API Syntax:

AmpImgDzoom_GetDzoomInfo (AMP_IMG_DZOOM_HDLR_s * ImgDzoom, AMP_IMG_DZOOM_INFO_s * Info)

Function Description:

- This function is used to get the current Dzoom information.

Parameters:

Type	Parameter	Description
AMP_IMG_DZOOM_HDLR_s *	**ImgDzoom**	ImgDzoom. (**AMP_IMG_DZOOM_HDLR_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.1.1 for more details.
AMP_IMG_DZOOM_INFO_s *	**Info**	Dzoom Information. (**AMP_IMG_DZOOM_INFO_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.5.1 for more details.

*Table 4-12. Parameters for ImgDzoom SDK6 API **AmpImgDzoom_GetDzoomInfo()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-13. Returns for ImgDzoom SDK6 API **AmpImgDzoom_GetDzoomInfo()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4.2.5.1 AmpIngDzoom_GetDzoomInfo > AMP_IMG_DZOOM_INFO_s

Type	Field	Description
UINT32	**TotalStepNumber**	Total step number of the dzoom
UINT32	**MaxDzoomFactor**	The maximum dzoom factor (16.16 format)

*Table 4-14. Definition of **AMP_IMG_DZOOM_INFO_s** for ImgDzoom SDK6 API **AmpImgDzoom_GetDzoomInfo()**.*

4.2.6 AmpImgDzoom_GetDzoomStatus

API Syntax:

AmpImgDzoom_GetDzoomStatus (AMP_IMG_DZOOM_HDLR_s * ImgDzoom, AMP_IMG_DZOOM_STATUS_s * Status)

Function Description:

- This function is to get the current ImageDzoom Status.

Parameters:

Type	Parameter	Description
AMP_IMG_DZOOM_HDLR_s *	**ImgDzoom**	ImgDzoom. (**AMP_IMG_DZOOM_HDLR_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.1.1 for more details.
AMP_IMG_DZOOM_STATUS_s *	**Status**	Dzoom Status. (**AMP_IMG_DZOOM_STATUS_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.6.1 for more details.

*Table 4-15. Parameters for ImgDzoom SDK6 API **AmpImgDzoom_GetDzoomStatus()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-16. Returns for ImgDzoom SDK6 API **AmpImgDzoom_GetDzoomStatus()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4.2.6.1 AmpImgDzoom_GetDzoomStatus > AMP_IMG_DZOOM_STATUS_s

Type	Field	Description
UINT32	**Region**	0: OPT_ZOOM_REGION 1: DIGI_ZOOM_REGION
UINT32	**IsProcessing**	0: DZOOM_NOTPROC 1: DZOOM_PROC
UINT32	**Step**	The current dzoom step
UINT32	**Factor**	Dzoom status (16.16 format)
int	**ShiftX**	Parameter of horizontal offset. Range: -1~1. '0': horizontal center of image. '-1': leftmost of horizontal image. '1': rightmost of horizontal image. (format: 16.16)
int	**ShiftY**	Parameter of vertical offset. Range: -1~1. '0': vertical center of the image. '-1': top of vertical image. '1': bottom of vertical image. (format: 16.16)

*Table 4-17. Definition of **AMP_IMG_DZOOM_STATUS_s** for ImgDzoom SDK6 API **AmpImgDzoom_GetDzoomStatus()**.*

4.2.7 AmpImgDzoom_GetInitDefaultCfg

API Syntax:

AmpImgDzoom_GetInitDefaultCfg (AMP_IMG_DZOOM_INIT_CFG_s * DefaultCfg)

Function Description:

- This function is to get ImgDzoom module default config for initializing.

Parameters:

Type	Parameter	Description
AMP_IMG_ DZOOM_INIT_ CFG_s *	**DefaultCfg**	ImgDzoom module config. (**AMP_IMG_DZOOM_INIT_CFG_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.7.1 for more details.

*Table 4-18. Parameters for ImgDzoom SDK6 API **AmpImgDzoom_GetInitDefaultCfg()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-19. Returns for ImgDzoom SDK6 API **AmpImgDzoom_GetInitDefaultCfg()**.*

Example

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4.2.7.1 AmpImgDzoom_GetInitDefaultCfg > AMP_IMG_DZOOM_INIT_CFG_s

Type	Field	Description
int	**NumOfMaxDzoom**	The number of maximum Dzoom

*Table 4-20. Definition of **AMP_IMG_DZOOM_INIT_CFG_s** for ImgDzoom SDK6 API **AmpImgDzoom_GetInitDefaultCfg()**.*

4.2.8 AmpImgDzoom_Init

API Syntax:

AmpImgDzoom_Init (AMP_IMG_DZOOM_INIT_CFG_s * Cfg)

Function Description:

- This function is to initialize ImgDzoom module.
- This function should only be invoked once. User MUST invoke this function before using ImgDzoom module. The memory pool of the module will be provided by the user.

Parameters:

Type	Parameter	Description
AMP_IMG_ DZOOM_INIT_ CFG_s *	**cfg**	ImgDzoom module config. (**AMP_IMG_DZOOM_INIT_CFG_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.7.1 for more details.

Table 4-21. Parameters for ImgDzoom SDK6 API ***AmpImgDzoom_Init()****.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 4-22. Returns for ImgDzoom SDK6 API ***AmpImgDzoom_Init()****.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4.2.9 AmpImgDzoom_PresetDzoomJump

API Syntax:

AmpImgDzoom_PresetDzoomJump (AMP_IMG_DZOOM_HDLR_s * ImgDzoom, AMP_IMG_DZOOM_JUMP_s * Jump)

Function Description:

- This function is to preset Dzoom Jump for ImgDzoom. Jump information will take effect after mode switch done.

Parameters:

Type	Parameter	Description
AMP_IMG_DZOOM_HDLR_s *	**ImgDzoom**	ImgDzoom. (**AMP_IMG_DZOOM_HDLR_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.1.1 for more details.
AMP_IMG_DZOOM_JUMP_s *	**Jump**	Dzoom Jump. (**AMP_IMG_DZOOM_JUMP_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.9.1 below for more details.

*Table 4-23. Parameters for SDK6 API **AmpImgDzoom_PresetDzoomJump()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-24. Returns for SDK6 API **AmpImgDzoom_PresetDzoomJump()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 21 for more details on error codes.

4.2.9.1 AmpImgDzoom_PresetDzoomJump > AMP_IMG_DZOOM_JUMP_s

Type	Field	Description
UINT32	**Step**	Step correspond to the dzoom table index
int	**ShiftX**	Parameter of horizontal offset. Range: -1~1. '0' : horizontal center of image. '-1': leftmost of horizontal image. '1': rightmost of horizontal image. (format: s16.16)
int	**ShiftY**	Parameter of vertical offset. Range: -1~1. '0': vertical center of image. '-1': top of vertical image. '1': bottom of vertical image. (format: s16.16)

*Table 4-25. Definition of **AMP_IMG_DZOOM_JUMP_s** for SDK6 API **AmpImgDzoom_PresetDzoomJump()**.*

4.2.10 AmpImgDzoom_RegDzoomTable

API Syntax:

AmpImgDzoom_RegDzoomTable (AMP_IMG_DZOOM_HDLR_s * ImgDzoom, AMP_IMG_DZOOM_TABLE_s * Table)

Function Description:

- This function is to register the Dzoom table for ImgDzoom.

Parameters:

Type	Parameter	Description
AMP_IMG_DZOOM_HDLR_s *	**ImgDzoom**	ImgDzoom. (**AMP_IMG_DZOOM_HDLR_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.1.1 for more details.
AMP_IMG_DZOOM_TABLE_s *	**Table**	Dzoom Table. (**AMP_IMG_DZOOM_TABLE_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.10.1 for more details.

*Table 4-26. Parameters for ImDzoom SDK6 API **AmpImgDzoom_RegDzoomTable()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-27. Returns for ImgDzoom SDK6 API **AmpImgDzoom_RegDzoomTable()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4.2.10.1 AmpImgDzoom_RegDzoomTable > AMP_IMG_DZOOM_TABLE_s

Type	Field	Description
UINT32	**TotalStepNumber**	Total step number of the dzoom
UINT32	**MaxDzoomFactor**	The maximum dzoom factor (16.16 format)
int(* GetDzoom-Factor)	**(int Step, UINT32 *Factor)**	Get the dzoom factor under corresponding status.

*Table 4-28. Definition of **AMP_IMG_DZOOM_TABLE_s** for ImDzoom SDK6 API **AmpImgDzoom_RegDzoomTable()**.*

4.2.11 AmpImgDzoom_ResetStatus

API Syntax:

AmpImgDzoom_ResetStatus (AMP_IMG_DZOOM_HDLR_s * ImgDzoom)

Function Description:

- This function is to reset the status in a ImgDzoom.

Parameters:

Type	Parameter	Description
AMP_IMG_ DZOOM_ HDLR_s *	**ImgDzoom**	ImgDzoom to reset the status. (**AMP_IMG_DZOOM_HDLR_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.1.1 for more details.

*Table 4-29. Parameters for ImgDzoom SDK6 API **AmpImgDzoom_ResetStatus()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-30. Returns for ImgDzoom SDK6 API **AmpImgDzoom_ResetStatus()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4.2.12 AmpImgDzoom_SetDzoomJump

API Syntax:

AmpImgDzoom_SetDzoomJump (AMP_IMG_DZOOM_HDLR_s * ImgDzoom, AMP_IMG_DZOOM_JUMP_s * Jump)

Function Description:

- This function is to set Dzoom Jump for ImgDzoom.

Parameters:

Type	Parameter	Description
AMP_IMG_DZOOM_HDLR_s *	**ImgDzoom**	ImgDzoom. (**AMP_IMG_DZOOM_HDLR_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.1.1 for more details.
AMP_IMG_DZOOM_JUMP_s *	**Jump**	Dzoom Jump. (**AMP_IMG_DZOOM_JUMP_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.9.1 for more details.

*Table 4-31. Parameters for ImgDzoom SDK6 API **AmpImgDzoom_SetDzoomJump()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-32. Returns for ImgDzoom SDK6 API **AmpImgDzoom_SetDzoomJump()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4.2.13 AmpImgDzoom_SetDzoomPosition

API Syntax:

AmpImgDzoom_SetDzoomPosition (AMP_IMG_DZOOM_HDLR_s * ImgDzoom, AMP_IMG_DZOOM_POSITION_s * Position)

Function Description:

- This function is to set Dzoom Position for ImgDzoom.

Parameters:

Type	Parameter	Description
AMP_IMG_DZOOM_HDLR_s *	**ImgDzoom**	ImgDzoom. (**AMP_IMG_DZOOM_HDLR_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.1.1 for more details.
AMP_IMG_DZOOM_POSITION_s *	**Position**	Dzoom Position. (**AMP_IMG_DZOOM_POSITION_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.13.1 for more details.

*Table 4-33. Parameters for ImgDzoom SDK6 API **AmpImgDzoom_SetDzoomPosition()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-34. Returns forImgDzoomSDK6 API **AmpImgDzoom_SetDzoomPosition()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4.2.13.1 AMP_IMG_DZOOM_POSITION_s

Type	Field	Description
UINT32	**Direction**	Direction for set dzoom position
UINT32	**Speed**	Speed for set dzoom position
UINT32	**Distance**	Distance for set dzoom position
int	**ShiftX**	Shift X for set dzoom position (16.16 format)
int	**ShiftY**	Shift Y for set dzoom position (16.16 format)

*Table 4-35. Definition of **AMP_IMG_DZOOM_POSITION_s** for ImgDzoom SDK6 API **AmpImgDzoom_SetDzoomPosition()**.*

4.2.14 AmpImgDzoom_StopDzoom

API Syntax:

AmpImgDzoom_StopDzoom (MP_IMG_DZOOM_HDLR_s * ImgDzoom)

Function Description:

- This function is to stop ImgDzoom. It will stop ImgDzoom during Moving or Jumping for ImgDzoom.

Parameters:

Type	Parameter	Description
AMP_IMG_ DZOOM_ HDLR_s *	**imgDzoomin**	ImgDzoom. (**AMP_IMG_DZOOM_HDLR_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.1.1 for more details.

*Table 4-36. Parameters for ImgDzoom SDK6 API **AmpImgDzoom_StopDzoom()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-37. Returns for ImgDzoom SDK6 API **AmpImgDzoom_StopDzoom()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4.2.15 AmpImgDzoom_Enable

API Syntax:

AmpImgDzoom_Enable (AMP_IMG_DZOOM_HDLR_s * ImgDzoom, UINT32 Enable)

Function Description:

- This function is used to enable/disable dzoom.

Parameters:

Type	Parameter	Description
AMP_IMG_ DZOOM_ HDLR_s *	**imgDzoomin**	ImgDzoom. (**AMP_IMG_DZOOM_HDLR_s** is defined in `ImgDzoom.h`) Please refer to Section 4.2.1.1 for more details.
UINT32	**Enable**	0: Disable dzoom function 1: Enable dzoom function

*Table 4-38. Parameters for ImgDzoom SDK6 API **AmpImgDzoom_Enable()**.*

Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 4-39. Returns for ImgDzoom SDK6 API **AmpImgDzoom_Enable()**.*

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

4.2.16 AmpImgDzoom_MainTask

API Syntax:

AmpImgDzoom_MainTask (UINT32 Input)

Function Description:

- This function is used to enable/disable dzoom.

Parameters:

Type	Parameter	Description
UINT32	**Input**	The task identification

*Table 4-40. Parameters for ImgDzoom SDK6 API **AmpImgDzoom_MainTask()**.*

Returns:

None

Example:

Please refer to Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

5 Initialization

5.1 Initialization: Overview

This chapter provides details on how to initialize the middleware of SDK6. It provides middleware level APIs for SDK6. The middleware utilizes system support package (SSP) with more easy-to-use APIs so that customers can develop product software's quickly and with ease. All initialization APIs are covered under this chapter as well.

5.2 Initialization: List of APIs

This section lists the following initialization APIs:

- AmpMW_GetDefaultInitCfg
- AmpMW_Init
- AmpResource_SetDspWorkArea
- AmpMW_GetFlowPipeVer
- AmpMW_GetDataPipeVer

5.2.1 AmpMW_GetDefaultInitCfg

API Syntax:

AmpMW_GetDefaultInitCfg (AMP_MW_INIT_CFG_s * cfg)

Function Description:

- This function gets the initial default configuration of the middleware.

Parameters:

Type	Parameter	Description
AMP_MW_INIT_CFG_s	**cfg**	Default built-in intialization configuration (**AMP_MW_INIT_CFG_s** is defined in `mw.h`) Please refer to Section 5.2.1.1 for more details.

*Table 5-1. Parameters for Initialization SDK6 API **AmpMW_GetDefaultInitCfg()**.*

Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 5-2. Returns for Initialization SDK6 API **AmpMW_GetDefaultInitCfg()**.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

5.2.1.1 AmpMW_GetDefaultInitCfg > AMP_MW_INIT_CFG_s

Type	Field	Description
UINT8 *	DspDbgLogDataArea	DSP debug message buffer start address
UINT32	SizeDspDbgLogDataArea	DSP debug message buffer size
UINT8 *	DspWorkingArea	DSP working buffer start address
UINT32	SizeDspWorkArea	DSP working buffer size
UINT8 *	ImgKernelWorkArea	Image kernel working buffer start address
UINT32	SizeImgKernelWorkArea	Image kernel working buffer size
UINT16	NumImgKernelPipe	Image kernel pipeline number
UINT16	NumImgKernelBufNum	Image kernel ring buffer number. Will get from ImgKernelWorkArea
AMBA_DSP_IMG_PIPE_INFO_s *	ImgKernelPipeInfo	Image kernel pipeline configuration

Type	Field	Description
UINT8	EnSeamless	If enable seamless function

*Table 5-3. Definition of **AMP_MW_INIT_CFG_s** for Initialization SDK6 API **AmpMW_GetDefaultInitCfg**().*

5.2.2 AmpMW_Init

API Syntax:

AmpMW_Init (AMP_MW_INIT_CFG_s * cfg)

Function Description:

- This function initializes the middleware.

Parameters:

Type	Parameter	Description
AMP_MW_INIT_ CFG_s	**cfg**	Default built-in initialization configuration (**AMP_MW_INIT_ CFG_s** is defined in `mw.h`). Please refer to Section 5.2.1.1 for more details.

*Table 5-4. Parameters for Initialization SDK6 API **AmpMW_Init()**.*

Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes

*Table 5-5. Returns for Initialization SDK6 API **AmpMW_Init()**.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

5.2.3 AmpMW_GetFlowPipeVer

API Syntax:

AmpMW_GetFlowPipeVer (void)

Function Description:

- This function gets the flow pipe SVN version.

Returns:

Return	Description
Version	SVN version number

Table 5-6. Returns for Initialization SDK6 API ***AmpMW_GetFlowPipeVer()****.*

Example:

Please refer to the Unit Test document.

See Also:

None

5.2.4 AmpMW_GetDataPipeVer

API Syntax:

AmpMW_GetDataPipeVer (void)

Function Description:

- This function gets the data pipe SVN version.

Returns:

Return	Description
Version	SVN version number

Table 5-7. Returns for Initialization SDK6 API ***AmpMW_GetDataPipeVer()****.*

Example:

Please refer to the Unit Test document.

See Also:

None

6 Flow Pipeline

6.1 Flow Pipeline: Overview

This chapter provides details on the encode / decode flow pipelines. Flow pipeline manages a group of codecs which take action together and share the same state.

6.2 Flow Pipeline: List of APIs for DecodeMgr Pipeline

This section lists the decoder pipeline API functions. The decode pipe is designed for codecs that need to be operated with time sync. In most cases, it is used for video decoding sync audio and video playback.

- AMPDec_GetDefaultCfg
- AMPDec_Create
- AMPDec_Delete
- AMPDec_Add
- AMPDec_Remove
- AMPDec_Start
- AMPDec_Pause
- AMPDec_Resume
- AMPDec_StopWithLastFrm
- AMPDec_Stop
- AMPDec_Step
- AMPDec_GetInfo

6.2.1 AMPDec_GetDefaultCfg

API Syntax:

AmpDec_GetDefaultCfg (AMP_DEC_PIPE_CFG_s * cfg)

Function Description:

- This function gets the default configuration of the decode manager.

Parameters:

Type	Parameter	Description
AMP_DEC_ PIPE_CFG_s*	**cfg**	Configuration of the decode manager (**AMP_DEC_PIPE_ CFG_s** is defined in `Decode.h`) Please refer to Section 6.2.1.1 below for more details.

Table 6-1. Parameters for Flow Pipeline SDK6 API **AmpDec_GetDefaultCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-2. Returns for Flow Pipeline SDK6 API **AmpDec_GetDefaultCfg()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.2.1.1 AmpDec_GetDefaultCfg > AMP_DEC_PIPE_CFG_s

Type	Field	Description
UINT32	**NumDecoder**	Number of handler(s) in the pipe
void*	**Decoder [MAX_DEC_PIPE_CO-DEC]**	Pointer to handler(s) in the pipe
AMP_ CALLBACK_f	**CbEvent**	Video decode pipe call back function

*Table 6-3. Definition of **AMP_DEC_PIPE_CFG_s** for Flow Pipeline SDK6 API **AmpDec_GetDefaultCfg**().*

6.2.2 AMPDec_Create

API Syntax:

AmpDec_Create (AMP_DEC_PIPE_CFG_s * cfg, AMP_DEC_PIPE_HDLR_s ** pipe)

Function Description:

- This function creates the decode manager.

Parameters:

Type	Parameter	Description
AMP_DEC_PIPE_CFG_s*	**cfg**	Configuration of the decode manager (**AMP_DEC_PIPE_CFG_s** is defined in `Decode.h`) Please refer to Section 6.2.1.1 for more details.
AMP_DEC_PIPE_HDLR_s**	**pipe**	Output pipe handler. (**AMP_DEC_PIPE_HDLR_s** is defined in `Decode.h`) Please refer to Section 6.2.2.1 for more details.

Table 6-4. Parameters for Flow Pipeline SDK6 API **AmpDec_Create()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-5. Returns for Flow Pipeline SDK6 API **AmpDec_Create()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.2.2.1 AmpDec_Create> AMP_DEC_PIPE_HDLR_s

Type	Field	Description
void*	**Ctx**	The context of the pipe.

*Table 6-6. Definition of **AMP_DEC_PIPE_HDLR_s** for Flow Pipeline SDK6 API **AmpDec_Create**().*

6.2.3 AMPDec_Delete

API Syntax:

AmpDec_Delete (AMP_DEC_PIPE_HDLR_s * pipe)

Function Description:

- This function deletes a pipe handler.

Parameters:

Type	Parameter	Description
AMP_DEC_ PIPE_HDLR_s*	**pipe**	The pipe to delete. (**AMP_DEC_PIPE_HDLR_s** is defined in `Decode.h`) Please refer to Section 6.2.2.1 for more details.

Table 6-7. Parameters for Flow Pipeline SDK6 API **AmpDec_Delete()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-8. Returns for Flow Pipeline SDK6 API **AmpDec_Delete()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.2.4 AMPDec_Add

API Syntax:

AmpDec_Add (AMP_DEC_PIPE_HDLR_s * pipe)

Function Description:

- This function adds a pipe to the decode manager.

Parameters:

Type	Parameter	Description
AMP_DEC_ PIPE_HDLR_s*	**pipe**	The pipe to add. (**AMP_DEC_PIPE_HDLR_s** is defined in `Decode.h`) Please refer to Section 6.2.2.1 for more details.

Table 6-9. Parameters for Flow Pipeline SDK6 API **AmpDec_Add()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-10. Returns for Flow Pipeline SDK6 API **AmpDec_Add()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.2.5 AMPDec_Remove

API Syntax:

AmpDec_Remove (AMP_DEC_PIPE_HDLR_s * pipe)

Function Description:

- This function removes the pipe from the decode manager.

Parameters:

Type	Parameter	Description
AMP_DEC_ PIPE_HDLR_s*	**pipe**	The pipe to remove. (**AMP_DEC_PIPE_HDLR_s** is defined in `Decode.h`) Please refer to Section 6.2.2.1 for more details.

Table 6-11. Parameters for Flow Pipeline SDK6 API **AmpDec_Remove()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-12. Returns for Flow Pipeline SDK6 API **AmpDec_Remove()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.2.6 AMPDec_Start

API Syntax:

AmpDec_Start (AMP_DEC_PIPE_HDLR_s * pipe, AMP_AVDEC_TRICKPLAY_s * trick, AMP_VIDEODEC_DISPLAY_s * display)

Function Description:

- This function is used to start the sync pipe.

Parameters:

Type	Parameter	Description
AMP_DEC_PIPE_HDLR_s*	**pipe**	The pipe to remove. (**AMP_DEC_PIPE_HDLR_s** is defined in `Decode.h`) Please refer to Section 6.2.2.1 for more details.
AMP_AVDEC_TRICKPLAY_s*	**trick**	The trickplay status (**AMP_AVDEC_TRICKPLAY_s** is defined in `Decode.h`) Please refer to Section 6.2.6.1 for more details.
AMP_VIDEODEC_DISPLAY_s*	**display**	(**AMP_VIDEODEC_DISPLAY_s** is defined in `Decode.h`) Please refer to Section 6.2.6.2 for more details.

Table 6-13. Parameters for Flow Pipeline SDK6 API ***AMPDec_Start()****.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-14. Returns for Flow Pipeline SDK6 API ***AMPDec_Start()****.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.2.6.1 AmpDec_Start> AMP_AVDEC_TRICKPLAY_s

Type	Field	Description
UINT32	**TimeOffsetOfFirstFrame**	The first frame to push to stream in ms
AMP_AVDEC_PLAY_DIRECTION_e	**Direction**	The playback direction

Type	Field	Description
UINT32	**Speed**	The playback speed

Table 6-15. Definition of ***AMP_AVDEC_TRICKPLAY_s*** *for Flow Pipeline SDK6 API* ***AmpDec_Start****().*

6.2.6.2 AmpDec_Start> AMP_VIDEODEC_DISPLAY_s

Type	Field	Description
UINT16	**SrcWidth**	The video width
UINT16	**SrcHeight**	The video height
AMP_AREA_s	**AOI**	The area to display

Table 6-16. Definition of ***AMP_AVDEC_TRICKPLAY_s*** *for Flow Pipeline SDK6 API* ***AmpDec_Start****().*

6.2.7 AMPDec_Pause

API Syntax:

AmpDec_Pause (AMP_DEC_PIPE_HDLR_s * pipe)

Function Description:

- This function is used to pause the sync pipe.

Parameters:

Type	Parameter	Description
AMP_DEC_ PIPE_HDLR_s*	**pipe**	The pipe to remove. (**AMP_DEC_PIPE_HDLR_s** is defined in `Decode.h`) Please refer to Section 6.2.2.1 for more details.

Table 6-17. Parameters for Flow Pipeline SDK6 API **AMPDec_Pause()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-18. Returns for Flow Pipeline SDK6 API **AMPDec_Pause()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.2.8 AMPDec_Resume

API Syntax:

AmpDec_Resume (AMP_DEC_PIPE_HDLR_s * pipe)

Function Description:

- This function resumes the sync pipe feature.

Parameters:

Type	Parameter	Description
AMP_DEC_ PIPE_HDLR_s*	**pipe**	The pipe to remove. (**AMP_DEC_PIPE_HDLR_s** is defined in `Decode.h`) Please refer to Section 6.2.2.1 for more details.

Table 6-19. Parameters for Flow Pipeline SDK6 API **AMPDec_Resume()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-20. Returns for Flow Pipeline SDK6 API **AMPDec_Resume()***)*.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.2.9 AMPDec_StopWithLastFrm

API Syntax:

AmpDec_StopWithLastFrm (AMP_DEC_PIPE_HDLR_s * pipe)

Function Description:

- This function is used to stop the pipe and hold the last frame on-screen.

Parameters:

Type	Parameter	Description
AMP_DEC_ PIPE_HDLR_s*	**pipe**	The pipe to remove. (**AMP_DEC_PIPE_HDLR_s** is defined in `Decode.h`) Please refer to Section 6.2.2.1 for more details.

Table 6-21. Parameters for Flow Pipeline SDK6 API **AMPDec_StopWithLastFrm()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-22. Returns for Flow Pipeline SDK6 API **AMPDec_StopWithLastFrm()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.2.10 AMPDec_Stop

API Syntax:

AmpDec_Stop (AMP_DEC_PIPE_HDLR_s * pipe)

Function Description:

- This function is used to stop the pipe.

Parameters:

Type	Parameter	Description
AMP_DEC_ PIPE_HDLR_s*	**pipe**	The pipe to remove. (**AMP_DEC_PIPE_HDLR_s** is defined in `Decode.h`) Please refer to Section 6.2.2.1 for more details.

Table 6-23. Parameters for Flow Pipeline SDK6 API **AMPDec_Stop()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-24. Returns for Flow Pipeline SDK6 API **AMPDec_Stop()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.2.11 AMPDec_Step

API Syntax:

AmpDec_Step (AMP_DEC_PIPE_HDLR_s * pipe)

Function Description:

- This function is used to make one step forward on the pipe.

Parameters:

Type	Parameter	Description
AMP_DEC_ PIPE_HDLR_s*	**pipe**	The pipe to remove. (**AMP_DEC_PIPE_HDLR_s** is defined in `Decode.h`) Please refer to Section 6.2.2.1 for more details.

Table 6-25. Parameters for Flow Pipeline SDK6 API **AMPDec_Step()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-26. Returns for Flow Pipeline SDK6 API **AMPDec_Step()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.2.12 AMPDec_GetInfo

API Syntax:

AmpDec_GetInfo (AMP_DEC_PIPE_HDLR_s * pipe, AMP_DEC_PIPE_INFO_s * info)

Function Description:

- This function is used to decode manager information.

Parameters:

Type	Parameter	Description
AMP_DEC_ PIPE_HDLR_s*	**pipe**	The pipe to remove. (**AMP_DEC_PIPE_HDLR_s** is defined in `Decode.h`) Please refer to Section 6.2.2.1 for more details.
AMP_DEC_ PIPE_INFO_s*	**pipe**	The pipe to remove. (**AMP_DEC_PIPE_INFO_s** is defined in `Decode.h`) Please refer to Section 6.2.12.1 for more details.

Table 6-27. Parameters for Flow Pipeline SDK6 API ***AMPDec_GetInfo()****.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-28. Returns for Flow Pipeline SDK6 API ***AMPDec_GetInfo()***

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.2.12.1 AmpDec_GetInfo> AMP_DEC_PIPE_INFO_s

Type	Field	Description
AMP_DEC_ PIPE_STATE_e	**State**	

Table 6-29. Definition of ***AMP_DEC_PIPE_INFO_s*** *for Flow Pipeline SDK6 API* ***AmpDec_GetInfo****().*

6.3 Flow Pipeline: List of APIs for EncodeMgr Pipeline

This section lists the APIs for the encode pipeline module implementation. The APIs are used to create encode, delete, add, remove encode pipeline module, start/stop liveview, start/stop/pause/resume encode, run/stop still capture, and get encode pipeline module information.

- AmpEnc_GetDefaultCfg
- AmpEnc_Create
- AmpEnc_Delete
- AmpEnc_Add
- AmpEnc_Remove
- AmpEnc_StartLiveview
- AmpEnc_StopLiveview
- AmpEnc_RunScript
- AmpEnc_StopRunScript
- AmpEnc_StartRecord
- AmpEnc_PauseRecord
- AmcEnc_ResumeRecord
- AmcEnc_StopRecord
- AmpEnc_GetInfo

6.3.1 AmpEnc_GetDefaultCfg

API Syntax:

AmpEnc_GetDefaultCfg (AMP_ENC_PIPE_CFG_s * cfg)

Function Description:

- This function gets the default pipeline configuration.

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_CFG_S*	**cfg**	Pipeline Configuration. (**AMP_ENC_PIPE_CFG_s** is defined in `Encode.h`) Please refer to Section 6.3.1.1 for more details.

Table 6-30. Parameters for Flow Pipeline SDK6 API ***AMPEnc_GetDefaultCfg()****.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-31. Returns for Flow Pipeline SDK6 API ***AMPEnc_GetDefaultCfg()****.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.1.1 AmpEnc_GetDefaultCfg> AMP_ENC_PIPE_CFG_s

Type	Field	Description
AMP_ENC_ PIPE_TYPE_e	**type**	Pipeline Type
UINT32	**numEncoder**	Number of handler(s) in the pipe
void*	**encoder[MAX_ENC_PIPE_CODEC]**	Pointer to handler(s) in the pipe
AMP_ CALLBACK_f	**cbEvent**	Callback function

Table 6-32. Definition of ***AMP_DEC_PIPE_CFG_s*** *for Flow Pipeline SDK6 API* ***AmpEnc_GetDefaultCfg****().*

6.3.2 AmpEnc_Create

API Syntax:

AmpEnc_Create (AMP_ENC_PIPE_CFG_s * cfg, AMP_ENC_PIPE_HDLR_s ** pipe)

Function Description:

- This function is used to create an encoder pipeline and set the initial configuration.

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_CFG_S*	**cfg**	Pipeline Configuration. (**AMP_ENC_PIPE_CFG_s** is defined in `Encode.h`) Please refer to Section 6.3.1.1 for more details.
AMP_ENC_ PIPE_HDLR_S*	**pipe**	Pipeline Handler. (**AMP_ENC_PIPE_HDLR_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.

Table 6-33. Parameters for Flow Pipeline SDK6 API ***AMPEnc_Create()****.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-34. Returns for Flow Pipeline SDK6 API ***AMPEnc_Create()****.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.2.1 AmpEnc_Create> AMP_ENC_PIPE_HDLR_s

Type	Field	Description
void	**ctx**	Pipeline handler: Context of the pipe

Table 6-35. Definition of ***AMP_DEC_PIPE_CFG_s*** *for Flow Pipeline SDK6 API* ***AmpEnc_GetDefaultCfg****().*

6.3.3 AmpEnc_Delete

API Syntax:

AmpEnc_Delete (AMP_ENC_PIPE_HDLR_s * pipe)

Function Description:

- This function is used to delete an encoder pipeline.

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_HDLR_S*	**pipe**	Pipeline Handler. (**AMP_ENC_PIPE_HDLR_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.

Table 6-36. Parameters for Flow Pipeline SDK6 API **AMPEnc_Delete()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-37. Returns for Flow Pipeline SDK6 API **AMPEnc_Delete()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.4 AmpEnc_Add

API Syntax:

AmpEnc_Add (AMP_ENC_PIPE_HDLR_s * pipe)

Function Description:

- This function is used to add a pipeline to the encode manager.

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_HDLR_S*	**pipe**	Pipeline Handler. (**AMP_ENC_PIPE_HDLR_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.

Table 6-38. Parameters for Flow Pipeline SDK6 API **AMPEnc_Add()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-39. Returns for Flow Pipeline SDK6 API **AMPEnc_Add()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.5 AmpEnc_Remove

API Syntax:

AmpEnc_Remove (AMP_ENC_PIPE_HDLR_s * pipe)

Function Description:

- This function is used to remove a pipeline from the encode manager.

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_HDLR_S*	**pipe**	Pipeline Handler. (**AMP_ENC_PIPE_HDLR_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.

Table 6-40. Parameters for Flow Pipeline SDK6 API **AMPEnc_Remove()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-41. Returns for Flow Pipeline SDK6 API **AMPEnc_Remove()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.6 AmpEnc_StartLiveview

API Syntax:

AmpEnc_StartLiveview (AMP_ENC_PIPE_HDLR_s * pipe, UINT32 flag)

Function Description:

- This function is used to ask a pipeline to start liveview.

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_HDLR_S*	**pipe**	Pipeline Handler. (**AMP_ENC_PIPE_HDLR_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.
UINT32	**flag**	/* No effect */ AMP_ENC_FUNC_FLAG_NONE /* blocked operation */ AMP_ENC_FUNC_FLAG_WAIT /* do not show previous video frame */ AMP_ENC_FUNC_FLAG_NO_SHOW_VIDEO_PLANE /* enter DSP suspend mode when stop liveview */ AMP_ENC_FUNC_FLAG_SUS_DSP_ON_STOP

Table 6-42. Parameters for Flow Pipeline SDK6 API **AMPEnc_StartLiveview()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-43. Returns for Flow Pipeline SDK6 API **AMPEnc_StartLiveview()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.7 AmpEnc_StopLiveview

API Syntax:

AmpEnc_StopLiveview (AMP_ENC_PIPE_HDLR_s * pipe, UINT32 flag)

Function Description:

- This function is used to ask a pipeline to stop liveview.

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_HDLR_S*	**pipe**	Pipeline Handler. (**AMP_ENC_PIPE_HDLR_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.

Table 6-44. Parameters for Flow Pipeline SDK6 API **AMPEnc_StopLiveview()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-45. Returns for Flow Pipeline SDK6 API **AMPEnc_StopLiveview()**.

Example

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.8 AmpEnc_RunScript

API Syntax:

AmpEnc_RunScript (AMP_ENC_PIPE_HDLR_s * pipe, void * script)

Function Description:

- This function is used to ask a pipeline to run script (still encode operation)

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_HDLR_S*	**pipe**	Pipeline Handler. (**AMP_ENC_PIPE_HDLR_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.
void*	**script**	Script Address

Table 6-46. Parameters for Flow Pipeline SDK6 API **AMPEnc_RunScript()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-47. Returns for Flow Pipeline SDK6 API **AMPEnc_RunScript()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.9 AmpEnc_StopRunScript

API Syntax:

AmpEnc_StopRunScript (AMP_ENC_PIPE_HDLR_s * pipe, UINT32 rule, UINT32 scrptID)

Function Description:

- This function is used to ask a pipeline to stop script (still encode operation).

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_HDLR_S*	**pipe**	Pipeline Handler. (**AMP_ENC_PIPE_HDLR_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.
UINT32	**rule**	Stop rule /* stop running script at last stage */ AMP_ENC_STOP_SCRIPT_NONE /* stop running script right away, whole pending data will be discard */ AMP_ENC_STOP_SCRIPT_RIGHT_AWAY
UINT32	**scrptID**	Script unique ID

Table 6-48. Parameters for Flow Pipeline SDK6 API **AMPEnc_StopRunScript()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 6-49. Returns for Flow Pipeline SDK6 API **AMPEnc_StopRunScript()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.10 AmpEnc_StartRecord

API Syntax:

AmpEnc_StartRecord (AMP_ENC_PIPE_HDLR_s * pipe, UINT32 flag)

Function Description:

- This function is used to ask a pipeline to start record.

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_HDLR_S*	**pipe**	Pipeline Handler. (**AMP_ENC_PIPE_HDLR_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.
UINT32	**flag**	Please refer to the flag description.

Table 6-50. Parameters for Flow Pipeline SDK6 API **AMPEnc_StartRecord()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	`AMP_ER_CODE_e`. Please refer to Chapter 10 for more details on error codes.

Table 6-51. Returns for API Flow Pipeline SDK6 API **AMPEnc_StartRecord()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.11 AmpEnc_PauseRecord

API Syntax:

AmpEnc_PauseRecord (AMP_ENC_PIPE_HDLR_s * pipe, UINT32 flag)

Function Description:

- This function is used to ask a pipeline to pause record.

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_HDLR_S*	**pipe**	Pipeline Handler. (**AMP_ENC_PIPE_HDLR_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.
UINT32	**flag**	Please refer to the flag description

Table 6-52. Parameters for Flow Pipeline SDK6 API **AMPEnc_PauseRecord()**.

Returns:

Return	Description
0	Success
- 1	Failure

Table 6-53. Returns for Flow Pipeline SDK6 API **AMPEnc_PauseRecord()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.12 AmcEnc_ResumeRecord

API Syntax:

AmpEnc_ResumeRecord (AMP_ENC_PIPE_HDLR_s * pipe, UINT32 flag)

Function Description:

- This function is used to ask a pipeline to resume record.

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_HDLR_S*	**pipe**	Pipeline Handler. (**AMP_ENC_PIPE_HDLR_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.
UINT32	**flag**	Please refer to flag description.

Table 6-54. Parameters for Flow Pipeline SDK6 API **AMPEnc_ResumeRecord()**.

Returns:

Return	Description
0	Success
- 1	Failure

Table 6-55. Returns for Flow Pipeline SDK6 API **AMPEnc_ResumeRecord()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.13 AmcEnc_StopRecord

API Syntax:

AmpEnc_StopRecord (AMP_ENC_PIPE_HDLR_s * pipe, UINT32 flag)

Function Description:

- This function is used to ask a pipeline to stop record.

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_HDLR_S*	**pipe**	Pipeline Handler. (**AMP_ENC_PIPE_HDLR_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.
UINT32	**flag**	Please refer to flag description.

Table 6-56. Parameters for Flow Pipeline SDK6 API **AMPEnc_StopRecord()**.

Returns:

Return	Description
0	Success
- 1	Failure

Table 6-57. Returns for Flow Pipeline SDK6 API **AMPEnc_StopRecord()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.14 AmpEnc_GetInfo

API Syntax:

AmpEnc_GetInfo (AMP_ENC_PIPE_HDLR_s * pipe, AMP_ENC_PIPE_INFO_s * info)

Function Description:

- This function is used to get encode manager information.

Parameters:

Type	Parameter	Description
AMP_ENC_ PIPE_HDLR_S*	**pipe**	Pipeline Handler. (**AMP_ENC_PIPE_HDLR_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.
AMP_ENC_ PIPE_INFO_S*	**info**	Pipeline State (**AMP_ENC_PIPE_INFO_s** is defined in `Encode.h`) Please refer to Section 6.3.2.1 for more details.

Table 6-58. Parameters for Flow Pipeline SDK6 API ***AMPEnc_GetInfo()****.*

Returns:

Return	Description
0	Success
- 1	Failure

Table 6-59. Returns for Flow Pipeline SDK6 API ***AMPEnc_GetInfo()****.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

6.3.14.1 AmpEnc_GetInfo> AMP_ENC_PIPE_INFO_s

Type	Field	Description
AMP_ENC_ PIPE_STATE_e	**state**	Pipeline State *AMP_ENC_PIPE_STATE_IDLE Idle State* *AMP_ENC_PIPE_STATE_LIVEVIEW Liveview State* *AMP_ENC_PIPE_STATE_RECORDING A/V recording state* *AMP_ENC_PIPE_STATE_RECORDING_PAUSED... A/V record-pause state* *AMP_ENC_PIPE_STATE_SCRIPT_PROCESSING...Still encode processing state*

Type	Field	Description
AMP_ENC_ PIPE_TYPE_e	**type**	Pipeline Type *AMP_ENC_AV_PIPE* *Audio or video pipeline* *AMP_ENC_STILL_PIPE* *Still pipeline*

Table 6-60. Definition of ***AMP_DEC_PIPE_CFG_s*** *for Flow Pipeline SDK6 API* ***AmpEnc_GetDefaultCfg****().*

7 Codec

7.1 Codec: Overview

This chapter provides details on the encoder and decoder engines. This is the basic unit of encoder and decoder implementation. Each codec can perform a specific function such as video encoding, audio encoding or video decoding and audio decoding.

7.2 Codec: List of APIs

This section lists the Amba Audio decoder APIs. This section provides the Amba audio decoder implementation. The audio decode module provides functions on the audio decode. Currently, Ambarella provides two decoding modes; PURE_AUDIO and VIDEO_AUDIO. The module also provides functions that initialize audio output which should be invoked before decode.

- AmpAudioDec_GetInitDefaultCfg
- AmpAudioDec_Init
- AmpAudioDec_GetDefaultCfg
- AmpAudioDec_Create
- AmpAudioDec_Delete
- AmpAudioDec_DecoderCfg
- AmpAudio_OutputInit
- AmpDummyDec_GetDefaultCfg
- AmpDummyDec_Create
- AmpDummyDec_Delete
- AmpStillDec_GetInitDefaultCfg
- AmpStillDec_Init
- AmpStillDec_GetDefaultCfg
- AmpStillDec_Create
- AmpStillDec_Delete
- AmpStillDec_Decode
- AmpStillDec_Rescale
- AmpStillDec_Display
- AmpStillDec_Blend
- AmpVideoDec_GetInitDefaultCfg
- AmpVideoDec_Init
- AmpVideoDec_GetDefaultCfg
- AmpVideoDec_Create
- AmpVideoDec_Delete
- AmpVideoDec_DumpFrame
- AmpVideoDec_GetTime
- AmpVideoDec_SetEosPts

- AmpVideoDec_GetXCodec
- AmpVideoDec_SetFadingEffect
- AmpAudioEnc_GetInitDefaultCfg
- AmpAudioEnc_Init
- AmpAudioEnc_GetDefaultCfg
- AmpAudioEnc_Create
- AmpAudioEnc_Delete
- AmpAudioEnc_Config
- AmpDummyEnc_GetDefaultCfg
- AmpDummyEnc_Create
- AmpDummyEnc_Delete
- AmpStillEnc_GetInitDefaultCfg
- AmpStillEnc_Init
- AmpStillEnc_GetDefaultCfg
- AmpStillEnc_Create
- AmpStillEnc_Delete
- AmpStillEnc_ConfigVinMain
- AmpStillEnc_StartRawCapture
- AmpStillEnc_StartFollowingRawCapture
- AmpStillEnc_UpdateVinMain
- AmpStillEnc_EnableLiveviewCapture
- AmpVideoEnc_GetInitDefaultCfg
- AmpVideoEnc_Init
- AmpVideoEnc_GetDefaultCfg
- AmpVideoEnc_Create
- AmpVideoEnc_Delete
- AmpVideoEnc_ConfigVinMain
- AmpVideoEnc_GetLayout
- AmpVideoEnc_SetBitstreamConfig
- AmpVideoEnc_SetBitstreamBuffer
- AmpVideoEnc_GetBitstreamConfig
- AmpVideoEnc_GetDefaultH264Header
- AmpVideoEnc_SetRuntimeBitrate
- AmpVideoEnc_SetRuntimeQuality
- AmpVideoEnc_CaptureTimeLapsedFrame
- AmpVideoEnc_SetRuntimeFrameRate
- AmpVideoEnc_SetEncodeBlend
- AmpVideoEnc_FeedEncodeRaw
- AmpVideoEnc_GetEncodingInfo

7.2.1 AmpAudioDec_GetInitDefaultCfg

API Syntax:

AmpAudioDec_GetInitDefaultCfg (AMP_AUDIODEC_INIT_CFG_s * cfg)

Function Description:

- This function is used to get the default value for the initial configuration for the audio decode module.

Parameters:

Type	Parameter	Description
AMP_AUDIO-DEC_INIT_CFG_s	**cfg**	Default module configuration (**AMP_AUDIODEC_INIT_CFG_s** is defined in `AudioDec.h`) Please refer to Section 7.2.1.1 for more details.

Table 7-1. Parameters for Codec SDK6 API **AmpAudioDec_GetInitDefaultCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-2. Returns for Codec SDK6 API **AmpAudioDec_GetInitDefaultCfg()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.2.1.1 AmpAudioDec_GetinitDefaultCfg> AMP_AUDIODEC_INIT_CFG_s

Type	Field	Description
UINT32	**MaxHdlr**	Maximum audio decoder codec that the user can create
AMP_TASK_INFO_s	**TaskInfo**	Audio decode task information (**AMP_TASK_INFO_s** is defined in `common.h`) Please refer to Section 7.2.1.1.1 for more details.
UINT8	**WorkBuff**	Audio decode working buffer
UINT32	**WorkBuffSize**	Size of the audio decode working buffer

*Table 7-3. Definition of **AMP_AUDIODEC_INIT_CFG_s** for Codec SDK6 API **AmpAudioDec_GetInitDefaultCfg**().*

7.2.1.1.1 AMP_AUDIODEC_INIT_CFG_s > AMP_TASK_INFO_s

Type	Field	Description
UINT32	**Priority**	Task priority
UINT32	**StackSize**	Task stack size

*Table 7-4. Definition of **AMP_TASK_STREAM_INFO_s** for Codec SDK6 API **AmpAudioDec_GetInitDefaultCfg**().*

7.2.2 AmpAudioDec_Init

API Syntax:

AmpAudioDec_Init (AMP_AUDIODEC_INIT_CFG_s * cfg)

Function Description:

- This function is used to set the initial configuration for the audio decode module.

Parameters:

Type	Parameter	Description
AMP_AUDIO-DEC_INIT_CFG_s	**cfg**	Default module configuration (**AMP_AUDIODEC_INIT_CFG_s** is defined in `AudioDec.h`) Please refer to Section 7.2.1.1 for more details.

Table 7-5. Parameters for Codec SDK6 API **AmpAudioDec_Init()***.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-6. Returns for Codec SDK6 API **AmpAudioDec_Init()***.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.2.3 AmpAudioDec_GetDefaultCfg

API Syntax:

AmpAudioDec_GetDefaultCfg (AMP_AUDIODEC_HDLR_CFG_s * cfg)

Function Description:

- This function is used to get the audio decode codec default configuration.

Parameters:

Type	Parameter	Description
AMP_AUDIO-DEC_HDLR_CFG_s	**cfg**	Default codec configuration (**AMP_AUDIODEC_HDLR_CFG_s** is defined in `AudioDec.h`) Please refer to Section 7.2.3.1 for more details.

Table 7-7. Parameters for Codec SDK6 API **AmpAudioDec_GetDefaultCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-8. Returns for Codec SDK6 API **AmpAudioDec_GetDefaultCfg()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.2.3.1 AmpAudioDec_GetDefaultCfg> AMP_AUDIODEC_HDLR_CFG_s

Type	Field	Description
UINT8	**DecoderTaskPriority**	Audio decoder task priority
AMBA_AUDIO_DEC_FLOW_e	**PureAudio**	Codec for pure audio or video audio
AMBA_AUDIO_TYPE_e	**DecType**	Audio decoder type
UINT32	**FadeInTime**	Fade in when decode starts
UINT32	**FadeOutTime**	Fade out when decode stops
UINT32	**SrcSampleRate**	Sample frequency of the input
UINT32	**DstSampleRate**	Sample frequency of the output

Type	Field	Description
UINT16	**SrcChannelMode**	Channel mode of the input
UINT16	**DstChannelMode**	Channel mode of the output
UINT32	**MaxSampleRate**	Maximum supporting sample frequency
UINT32	**MaxChannelNum**	Maximum channel number
UINT32	**MaxFrameSize**	Maximum frame size
UINT32	**MaxChunkNum**	Maximum chunk number
UINT8	**I2SIndex**	I2S Index of output
UINT8*	**CodecCacheWorkBuff**	Buffer address for audio codec
UINT32	**CodecCacheWorkSize**	Buffer size for audio codec
union{AMBA_AUDIO_PCM_CONFIG_s PCMCfg, AMBA_AUDIO_AACDEC_CONFIG_s AACCfg}	**Spec**	Decode information
AMP_CALLBACK_f	**CbCfgUpdated**	Setting takes effect
AMP_CALLBACK_f	**CbEvent**	Event related to the encoder
UINT8*	**RawBuffer**	Buffer address for audio raw file
UINT32	**RawBufferSize**	Size of buffer address for audio raw file
AMP_BITS_DESC_s*	**DescBuffer**	Buffer address for audio descriptor
UINT32	**DescBufferNum**	Size of buffer address for audio descriptor

*Table 7-9. Definition of **AMP_AUDIODEC_HDLR_CFG_s** for Codec SDK6 API **AmpAudioDec_GetDefaultCfg**().*

7.2.4 AmpAudioDec_Create

API Syntax:

AmpAudioDec_Create (AMP_AUDIODEC_HDLR_CFG_s * cfg, AMP_AVDEC_HDLR_s ** hdlr)

Function Description:

- This function is used to create the audio decode codec.

Parameters:

Type	Parameter	Description
AMP_AUDIODEC_HDLR_CFG_s*	**cfg**	Default codec configuration (**AMP_AUDIODEC_HDLR_CFG_s** is defined in `AudioDec.h`) Please refer to Section 7.2.3.1 for more details.
AMP_AVDEC_HDLR_s**	**hdlr**	Codec handler structure for video decode (**AMP_AVDEC_HDLR_s** is defined in `Decode.h`) Please refer to Section 7.2.4.1 for more details.

Table 7-10. Parameters for Codec SDK6 API ***AmpAudioDec_Create()****.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-11. Returns for Codec SDK6 API ***AmpAudioDec_Create()****.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.2.4.1 AmpAudioDec_Create> AMP_AVDEC_HDLR_s

Type	Field	Description
void*	**Ctx**	Pointer to codec context
AMP_AVDEC_CODEC_s*	**Function**	Pointer to codec operation functions

Table 7-12. Definition of ***AMP_AVDEC_HDLR_s*** *for Codec SDK6 API* ***AmpAudioDec_Create****().*

7.2.5 AmpAudioDec_Delete

API Syntax:

AmpAudioDec_Delete (AMP_AVDEC_HDLR_s * hdlr)

Function Description:

- This function is used to delete the audio decode codec.

Parameters:

Type	Parameter	Description
AMP_AVDEC_ HDLR_s	**hdlr**	Codec handler structure for video decode (**AMP_AVDEC_ HDLR_s** is defined in `Decode.h`) Please refer to Section 7.2.4.1 for more details.

Table 7-13. Parameters for Codec SDK6 API **AmpAudioDec_Delete()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-14. Returns for Codec SDK6 API **AmpAudioDec_Delete()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.2.6 AmpAudioDec_DecoderCfg

API Syntax:

AmpAudioDec_DecoderCfg (AMP_AUDIODEC_DECODER_CFG_s *cfg, AMP_AVDEC_HDLR_s *hdlr)

Function Description:

- This function is used to configure the decoder parameters when in idle state.

Parameters:

Type	Parameter	Description
AMP_AUDIODEC_DECODER_CFG_s	**cfg**	Audio Decoder configuration (**AMP_AUDIODEC_DECODER_CFG_s** is defined in `AudioDec.h`) Please refer to Section 7.2.6.1 for more details.
AMP_AVDEC_HDLR_s	**hdlr**	Codec handler structure for video decode (**AMP_AVDEC_HDLR_s** is defined in `Decode.h`) Please refer to Section 7.2.4.1 for more details.

Table 7-15. Parameters for Codec SDK6 API **AmpAudioDec_DecoderCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-16. Returns for Codec SDK6 API **AmpAudioDec_DecoderCfg()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.2.6.1 AmpAudioDec_DecoderCfg> AMP_AUDIODEC_DECODER_CFG_s

Type	Field	Description
AMBA_AUDIO_TYPE_e	**DecType**	
UINT32	**FadeInTime**	Fade in when decode starts
UINT32	**FadeOutTime**	Fade out when decode stops
UINT32	**SrcSampleRate**	Sample frequency of the input
UINT16	**SrcChannelMode**	Channel mode of the input

Type	Field	Description
union{AMBA_AUDIO_PCM_CONFIG_s PCMCfg, AMBA_AUDIO_AACDEC_CONFIG_s AACCfg}	**Spec**	Decode information

*Table 7-17. Definition of **AMP_AVDEC_HDLR_s** for Codec SDK6 API **AmpAudioDec_Create**().*

7.2.7 AmpAudio_OutputInit

API Syntax:

AmpAudio_OutputInit (UINT32 * outputCtrl, UINT8 priority)

Function Description:

- This function is used to initialize the audio output task.

Parameters:

Type	Parameter	Description
UINT32	**outputCtrl**	Audio output resource
UINT8	**priority**	Task priority

Table 7-18. Parameters for Codec SDK6 API **AmpAudioDec_OutputInit()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-19. Returns for Codec SDK6 API **AmpAudioDec_OutputInit()**.

Example

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.3 Codec: List of APIs for DummyDec

This section lists the dummy decoder APIs. This section provides the dummy decoder function sets. Dummy decoder is a decoder that only consumes frames without doing anything. It is used for testing purposes only.

- AmpDummyDec_GetDefaultCfg
- AmpDummyDec_Create
- AmpDummyDec_Delete

7.3.1 AmpDummyDec_GetDefaultCfg

API Syntax:

AmpDummyDec_GetDefaultCfg (AMP_DUMMYDEC_CFG_s * cfg)

Function Description:

- This function is used to get the default configuration of a dummy video decode codec.

Parameters:

Type	Parameter	Description
AMP_DUMMYDEC_CFG_s	**cfg**	Codec configuration (**AMP_DUMMYDEC_CFG_s** is defined in `DummyDec.h`) Please refer to Section 7.3.1.1 for more details.

Table 7-20. Parameters for Codec SDK6 API **AmpDummyDec_GetDefaultCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-21. Returns for Codec SDK6 API **AmpDummyDec_GetDefaultCfg()***)*.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.3.1.1 AmpDummyDec_GetDefaultCfg> AMP_DUMMYDEC_CFG_s

Type	Field	Description
AMBA_YUV_COLOR_s	**BgColor**	Background color
AMP_CALLBACK_f	**CbCfgUpdated**	Configuration updated cb function
AMP_CALLBACK_f	**CbCodecEvent**	Codec state cb function
char	**RawBuffer**	Raw buffer for coded data
UINT32	**RawBufferSize**	Size of the raw buffer
char	**DescBuffer**	Desc buffer
UINT32	**NumDescBuffer**	Size of desc buffer

*Table 7-22. Definition of **AMP_DUMMYDEC_CFG_s** for Codec SDK6 API **AmpDummyDec_GetDefaultCfg**().*

7.3.2 AmpDummyDec_Create

API Syntax:

AmpDummyDec_Create (AMP_DUMMYDEC_CFG_s * cfg, AMP_AVDEC_HDLR_s ** hdlr)

Function Description:

- This function is used to create a dummy video decode codec.

Parameters:

Type	Parameter	Description
AMP_DUMMYDEC_CFG_s	**cfg**	Codec configuration (**AMP_DUMMYDEC_CFG_s** is defined in `DummyDec.h`) Please refer to Section 7.3.1.1 for more details.
AMP_AVDEC_HDLR_s	**hdlr**	Codec handler structure for video decode (**AMP_AVDEC_HDLR_s** is defined in `Decode.h`) Please refer to Section 7.2.4.1 for more details.

Table 7-23. Parameters for Codec SDK6 API **AmpDummyDec_Create()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-24. Returns for Codec SDK6 API **AmpDummyDec_Create()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.3.3 AmpDummyDec_Delete

API Syntax:

AmpDummyDec_Delete (AMP_AVDEC_HDLR_s * hdlr)

Function Description:

- This function is used to delete a dummy video decode codec.

Parameters:

Type	Parameter	Description
AMP_AVDEC_HDLR_s	**hdlr**	Codec handler structure for video decode (**AMP_AVDEC_HDLR_s** is defined in `Decode.h`) Please refer to Section 7.2.4.1 for more details.

Table 7-25. Parameters for Codec SDK6 API **AmpDummyDec_Delete()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-26. Returns for Codec SDK6 API **AmpDummyDec_Delete()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.4 Codec: List of APIs for StillDec

This section lists the still image decoder APIs. The still decoder codec provides functions that operate on a single frame. It includes Idr/I frame decode, JPEG decode, YUV rescale, YUV display, and YUV blending.

- AmpStillDec_GetInitDefaultCfg
- AmpStillDec_Init
- AmpStillDec_GetDefaultCfg
- AmpStillDec_Create
- AmpStillDec_Delete
- AmpStillDec_Decode
- AmpStillDec_Rescale
- AmpStillDec_Display
- AmpStillDec_Blend

7.4.1 AmpStillDec_GetInitDefaultCfg

API Syntax:

AmpStillDec_GetInitDefaultCfg (AMP_STILLDEC_INIT_CFG_s * cfg)

Function Description:

- This function is used to get the default value of initial configuration for the video decode module.

Parameters:

Type	Parameter	Description
AMP_STILLDEC_INIT_CFG_s	**cfg**	Default module configuration Please refer to Section 7.4.1.1 for more details.

Table 7-27. Parameters for Codec SDK6 API ***AmpStillDec_GetInitDefaultCfg()****.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-28. Returns for Codec SDK6 API ***AmpStillDec_GetInitDefaultCfg()****.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.4.1.1 AmpStillDec_GetInitDefaultCfg> AMP_STILLDEC_INIT_CFG_s

Type	Field	Description
UINT32	**MaxHdlr**	Max hdlr supported on the module
AMP_TASK_INFO_s	**TaskInfo**	Task info for decode task
UINT8*	**Buf**	Working buffer for still decode module
UINT32	**BufSize**	Working buffer size for still decode module

Table 7-29. Definition of ***AMP_STILLDEC_INIT_CFG_s*** *for Codec SDK6 API* ***AmpStillDec_GetInitDefaultCfg****().*

7.4.2 AmpStillDec_Init

API Syntax:

AmpStillDec_Init (AMP_STILLDEC_INIT_CFG_s * cfg)

Function Description:

- This function is used to set the initial configuration for the still decode module.

Parameters:

Type	Parameter	Description
AMP_STILL-DEC_INIT_CFG_s	**cfg**	Default module configuration Please refer to Section 7.4.1.1 for more details.

Table 7-30. Parameters for Codec SDK6 API **AmpStillDec_Init()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-31. Returns for Codec SDK6 API **AmpStillDec_Init()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.4.3 AmpStillDec_GetDefaultCfg

API Syntax:

AmpStillDec_GetDefaultCfg (AMP_STILLDEC_CFG_s * cfg)

Function Description:

- This function is used to get the default configuration of the still decode codec.

Parameters:

Type	Parameter	Description
AMP_STILLDEC_CFG_s	**cfg**	Default module configuration Please refer to Section 7.4.3.1 for more details.

Table 7-32. Parameters for Codec SDK6 API **AmpStillDec_GetDefaultCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-33. Returns for Codec SDK6 API **AmpStillDec_GetDefaultCfg()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.4.3.1 AmpStillDec_GetDefaultCfg> AMP_STILLDEC_CFG_s

Type	Field	Description
AMP_YUV_COLOR_s	**BgColor**	Background color
AMP_CALLBACK_f	**CbCodecEvent**	Callback function pointer of still decode event
UINT8*	**RawBuf**	Buffer for still raw file
UINT32	**RawBufSize**	Buffer size of the raw buffer

Table 7-34. Definition of ***AMP_STILLDEC_CFG_s*** *for Codec SDK6 API* ***AmpStillDec_GetDefaultCfg****().*

7.4.3.1.1 AmpStillDec_GetDefaultCfg> AMP_YUV_COLOR_s

Type	Field	Description
UINT8	**Y**	Y part of the color
UINT8	**U**	U part of the color
UINT8	**V**	V part of the color

*Table 7-35. Definition of **AMP_YUV_COLOR_s** for Codec SDK6 API **AmpStillDec_GetDefaultCfg**().*

7.4.4 AmpStillDec_Create

API Syntax:

AmpStillDec_Create (AMP_STILLDEC_CFG_s * cfg, AMP_STLDEC_HDLR_s ** hdlr)

Function Description:

- This function is used to create still decode codec.

Parameters:

Type	Parameter	Description
AMP_STILL-DEC_CFG_s	**cfg**	Default module configuration Please refer to Section 7.4.3.1 for more details.
AMP_STLDEC_HDLR_s	**hdlr**	Codec Handler. Please refer to Section 7.4.4.1 for more details

Table 7-36. Parameters for Codec SDK6 API **AmpStillDec_Create()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-37. Returns for Codec SDK6 API **AmpStillDec_Create()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.4.4.1 AmpStillDec_Create> AMP_STLDEC_HDLR_s

Type	Field	Description
void	**ctx**	Pointer to codec context
AMP_STLDEC_CODEC_s	**Function**	Pointer to codec operation functions

*Table 7-38. Definition of **AMP_STILLDEC_HDLR_s** for Codec SDK6 API **AmpStillDec_Create**().*

7.4.5 AmpStillDec_Delete

API Syntax:

AmpStillDec_Delete (AMP_STLDEC_HDLR_s * hdlr)

Function Description:

- This function deletes still decode codec.

Parameters:

Type	Parameter	Description
AMP_STLDEC_HDLR_s	**hdlr**	Codec Handler. Please refer to Section 7.4.4.1 for more details

Table 7-39. Parameters for Codec SDK6 API **AmpStillDec_Delete()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-40. Returns for Codec SDK6 API **AmpStillDec_Delete()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.4.6 AmpStillDec_Decode

API Syntax:

AmpStillDec_Decode (AMP_STLDEC_HDLR_s * hdlr, AMP_STILLDEC_DECODE_s * decode)

Function Description:

- This function is used to decode raw data into the YUV data.

Parameters:

Type	Parameter	Description
AMP_STLDEC_HDLR_s	**hdlr**	Codec Handler. Please refer to Section 7.4.4.1 for more details
AMP_STILLDEC_DECODE_s	**decode**	Please refer to Section 7.4.6.1 for more details.

Table 7-41. Parameters for Codec SDK6 API **AmpStillDec_Decode()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-42. Returns for Codec SDK6 API **AmpStillDec_Decode()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.4.6.1 AmpStillDec_Decode> AMP_STILLDEC_DECODE_s

Type	Field	Description
UINT32	**NumFile**	Number of file to decode
UINT8*	**DecodeBuf**	Decode output buffer
UINT32	**SizeDecodeBuf**	Size of decode output buffer
UINT32*	**DecodedImgPitch**	Pitch in decode buffer for last decoded image
UINT32*	**DecodeState**	U32 array whose size is numFile for store error state for each decode
AMP_AREA_s*	**CropFromDecodedBuf**	AMP_AREA_S array. Null to skip scaling.

Type	Field	Description
AMP_YUV_ BUFFER_s*	**RescaleDest**	AMP_YUV_BUFFER_s array
AMP_ ROTATION_e*	**Rotate**	Rotation on scaling
UINT8*	**OutAddrY**	The address of decoded Y data of LAST file
UINT8*	**OutAddrUV**	The address of decoded UV data of LAST file
AMP_COLOR_ FORMAT_e	**OutColorFmt**	The color format of LAST file

*Table 7-43. Definition of **AMP_STILLDEC_DECODE_s** for Codec SDK6 API **AmpStillDec_Decode**().*

7.4.7 AmpStillDec_Rescale

API Syntax:

AmpStillDec_Rescale (AMP_STLDEC_HDLR_s * hdlr, AMP_STILLDEC_RESCALE_s * rescale)

Function Description:

- This function is used to rescale a YUV buffer and the limitation is that it should be aligned with the value depending on the DSP hardware limitation.

Parameters:

Type	Parameter	Description
AMP_STLDEC_ HDLR_s	**hdlr**	Codec Handler. Please refer to Section 7.4.4.1 for more details
AMP_STILL-DEC_ RESCALE_s	**rescale**	Rescale Information. (**AMP_STILLDEC_RESCALE_s** is defined in `StillDec.h`) Please refer to Section 7.4.7.1 for more details.

Table 7-44. Parameters for Codec SDK6 API **AmpStillDec_Rescale()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-45. Returns for Codec SDK6 API **AmpStillDec_Rescale()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.4.7.1 AmpStillDec_Rescale> AMP_STILLDEC_RESCALE_s

Type	Field	Description
UINT32	**NumBuf**	Number of buffer to scale
AMP_YUV_ BUFFER_s	**Src**	Scaling source (**AMP_YUV_BUFFER_s** is defined in `common.h`) Please refer to Section 7.4.7.1.1 for more details.
AMP_YUV_ BUFFER_s	**Dest**	Scaling target (**AMP_YUV_BUFFER_s** is defined in `common.h`) Please refer to Section 7.4.7.1.1 for more details

Type	Field	Description
AMP_ ROTATION_e	**Rotate**	Rotate on scaling
UINT8	**LumaGain**	Luma gain: 1 ~ 255

*Table 7-46. Definition of **AMP_STILLDEC_Rescale_s** for Codec SDK6 API **AmpStillDec_Rescale**().*

7.4.7.1.1 AMP_STILLDEC_RESCALE_s > AMP_YUV_BUFFER_s

Type	Field	Description
AMP_COLOR_ FORMAT_e	**ColorFmt**	Color Format of the YUV buffer
UINT32	**Width**	Buffer width
UINT32	**Height**	Buffer height
UINT32	**Pitch**	Buffer pitch
UINT8	**LumaAddr**	Luma address
UINT8	**ChromaAddr**	Chroma address
AMP_AREA_s	**AOI**	Area of interest

*Table 7-47. Definition of **AMP_YUV_BUFFER_s** for Codec SDK6 API **AmpStillDec_Rescale**().*

7.4.8 AmpStillDec_Display

API Syntax:

AmpStillDec_Display (AMP_STLDEC_HDLR_s * hdlr, AMP_STILLDEC_DISPLAY_s * display)

Function Description:

- This function is used to display the buffer on Vout.

Parameters:

Type	Parameter	Description
AMP_STLDEC_HDLR_s	**hdlr**	Codec Handler. Please refer to Section 7.4.4.1 for more details
AMP_STILL-DEC_DISPLAY_s	**display**	Display Info (**AMP_STILLDEC_DISPLAY_s** is defined in `StillDec.h`) Please refer to Section 7.4.8.1 for more details.

Table 7-48. Parameters for Codec SDK6 API **AmpStillDec_Display()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-49. Returns for Codec SDK6 API **AmpStillDec_Display()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.4.8.1 AmpStillDec_Display> AMP_STILLDEC_DISPLAY_s

Type	Field	Description
AMP_DISP_CHANNEL_IDX_e	**Vout**	This function decides which Vout to display on
AMP_YUV_BUFFER_s	**Buf**	Buffer to display. Size of AOI must be the same as the Vout size. (**AMP_YUV_BUFFER_s** is defined in `common.h`) Please refer to Section 7.4.7.1.1 for more details

*Table 7-50. Definition of **AMP_STILLDEC_DISPLAY_s** for Codec SDK6 API **AmpStillDec_Display**().*

7.4.9 AmpStillDec_Blend

API Syntax:

AmpStillDec_Blend (AMP_STLDEC_HDLR_s *hdlr, AMP_STILLDEC_BLEND_s *blend)

Function Description:

- This function is used to blend two YUV buffers with the same height and pitch.

Parameters:

Type	Parameter	Description
AMP_STLDEC_HDLR_s	**hdlr**	Codec Handler. Please refer to Section 7.4.4.1 for more details
AMP_STILLDEC_BLEND_s	**blend**	Blending information. Please refer to Section 7.4.9.1 for more details

Table 7-51. Parameters for Codec SDK6 API **AmpStillDec_Blend()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-52. Returns for Codec SDK6 API **AmpStillDec_Blend()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.4.9.1 AmpStillDec_Blend> AMP_STILLDEC_BLEND_s

Type	Field	Description
AMP_YUV_BUFFER_s	**Src1**	Scaling source. (**AMP_YUV_BUFFER_s** is defined in `common.h`) Please refer to Section 7.4.7.1.1 for more details
AMP_YUV_BUFFER_s	**Src2**	Scaling source. (**AMP_YUV_BUFFER_s** is defined in `common.h`) Please refer to Section 7.4.7.1.1 for more details
AMP_YUV_BUFFER_s	**Dest**	Scaling target. (**AMP_YUV_BUFFER_s** is defined in `common.h`) Please refer to Section 7.4.7.1.1 for more details
UINT8	**AlphaMap**	Pointer to alpha map

Type	Field	Description
UINT32	**GlobalAlpha**	Take effect when alphaMap = NULL

*Table 7-53. Definition of **AMP_STILLDEC_BLEND_s** for Codec SDK6 API **AmpStillDec_Blend**().*

7.5 Codec: List of APIs for VideoDec

This section lists the Amba video decoder APIs. The video decode module provides functions on video decode. In most cases, the user controls video decode via the decode pipe. Some functions like GetInfo, SetEos, and DumpFrame can run directly.

- AmpVideoDec_GetInitDefaultCfg
- AmpVideoDec_Init
- AmpVideoDec_GetDefaultCfg
- AmpVideoDec_Create
- AmpVideoDec_Delete
- AmpVideoDec_DumpFrame
- AmpVideoDec_GetTime
- AmpVideoDec_SetEosPts
- AmpVideoDec_GetXCodec
- AmpVideoDec_SetFadingEffect

7.5.1 AmpVideoDec_GetInitDefaultCfg

API Syntax:

AmpVideoDec_GetInitDefaultCfg (AMP_VIDEODEC_INIT_CFG_s *cfg)

Function Description:

- This function is used to get the default value of the initial configuration for the video decode module.

Parameters:

Type	Parameter	Description
AMP_VIDEO-DEC_INIT_CFG_s	**cfg**	Default module configuration. (**AMP_VIDEODEC_INIT_CFG_s** is defined in `VideoDec.h`) Please refer to Section 7.5.1.1 for more details

Table 7-54. Parameters for Codec SDK6 API **AmpVideoDec_GetInitDefaultCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-55. Returns for Codec SDK6 API **AmpVideoDec_GetInitDefaultCfg()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.5.1.1 AmpVideoDec_GetInitDefaultCfg> AMP_VIDEODEC_INIT_CFG_s

Type	Field	Description
UINT32	**MaxHdlr**	
AMP_TASK_INFO_s	**TaskInfo**	(**AMP_TASK_INFO_s** is defined in `common.h`) Please refer to Section 7.5.1.1 for more details
UINT8	**Buf**	
UINT32	**BufSize**	

Table 7-56. Definition of ***AMP_VIDEODEC_INIT_CFG_s*** *for Codec SDK6 API* ***AmpVideoDec_GetInitDefaultCfg****()*.

7.5.1.1.1 AMP_VIDEODEC_INIT_CFG_s > AMP_TASK_INFO_s

Type	Field	Description
UINT32	**Priority**	Task Priority
UINT32	**StackSize**	Task stack size

*Table 7-57. Definition of **AMP_TASK_INFO_s** for Codec SDK6 API **AmpVideoDec_GetInitDefauitCfg**().*

7.5.2 AmpVideoDec_Init

API Syntax:

AmpVideoDec_Init (AMP_VIDEODEC_INIT_CFG_s * cfg)

Function Description:

- Thuis function is used to set the initial configuration for the video decode module.

Parameters:

Type	Parameter	Description
AMP_VIDEO-DEC_INIT_CFG_s	**cfg**	Default module configuration. (**AMP_VIDEODEC_INIT_CFG_s** is defined in `VideoDec.h`) Please refer to Section 7.5.1.1 for more details

Table 7-58. Parameters for Codec SDK6 API **AmpVideoDec_Init()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-59. Returns for Codec SDK6 API **AmpVideoDec_Init()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.5.3 AmpVideoDec_GetDefaultCfg

API Syntax:

AmpVideoDec_GetDefaultCfg (AMP_VIDEODEC_CFG_s * cfg)

Function Description:

- This function is used to get the video decode codec default configuration.

Parameters:

Type	Parameter	Description
AMP_VIDEODEC_CFG_s	**cfg**	Default codec configuration. (**AMP_VIDEODEC_CFG_s** is defined in `VideoDec.h`) Please refer to Section 7.5.3.1 for more details

Table 7-60. Parameters for Codec SDK6 API **AmpVideoDec_GetDefaultCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-61. Returns for Codec SDK6 API **AmpVideoDec_GetDefaultCfg()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.5.3.1 AmpVideoDec_GetDefaultCfg> AMP_VIDEODEC_CFG_s

Type	Field	Description
AMP_YUV_COLOR_s	**BgColor**	Background color
AMP_CALLBACK_f	**CbCfgUpdated**	Callback function when configuration is updated
AMP_CALLBACK_f	**CbCodecEvent**	Callback function on event
char	**RawBuffer**	Buffer for raw file as file reading buffer
UINT32	**RawBufferSize**	Size of raw buffer
char	**DescBuffer**	Buffer for internal descriptor
UINT32	**NumDescBuffer**	Number of descriptor

Type	Field	Description
AMP_VIDEODEC_FEATURE_CFG_s	**Feature**	Decode video initial configuration

*Table 7-62. Definition of **AMP_VIDEODEC_CFG_s** for Codec SDK6 API **AmpVideoDec_GetDefaultCfg**().*

7.5.4 AmpVideoDec_Create

API Syntax:

AmpVideoDec_Create (AMP_VIDEODEC_CFG_s * cfg, AMP_AVDEC_HDLR_s ** hdlr)

Function Description:

- This function is used to create the video decode codec.

Parameters:

Type	Parameter	Description
AMP_VIDEODEC_CFG_s	**cfg**	Default codec configuration. (**AMP_VIDEODEC_CFG_s** is defined in `VideoDec.h`) Please refer to Section 7.5.3.1 for more details
AMP_AVDEC_HDLR_s	**hdlr**	Codec handler structure for video decode (**AMP_AVDEC_HDLR_s** is defined in `Decode.h`) Please refer to Section 7.2.4.1 for more details.

Table 7-63. Parameters for Codec SDK6 API ***AmpVideoDec_Create()***.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-64. Returns for Codec SDK6 API ***AmpVideoDec_Create()***.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.5.5 AmpVideoDec_Delete

API Syntax:

AmpVideoDec_Delete (AMP_AVDEC_HDLR_s * hdlr)

Function Description:

- This function is used to delete the codec.

Parameters:

Type	Parameter	Description
AMP_AVDEC_HDLR_s	**hdlr**	Codec handler structure for video decode (**AMP_AVDEC_HDLR_s** is defined in `Decode.h`) Please refer to Section 7.2.4.1 for more details.

Table 7-65. Parameters for Codec SDK6 API **AmpVideoDec_Delete()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-66. Returns for Codec SDK6 API **AmpVideoDec_Delete()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.5.6 AmpVideoDec_DumpFrame

API Syntax:

AmpVideoDec_DumpFrame (AMP_AVDEC_HDLR_s * hdlr, AMP_YUV_BUFFER_s * buf)

Function Description:

-
-

Parameters:

Type	Parameter	Description
AMP_AVDEC_HDLR_s	**hdlr**	Codec handler structure for video decode (**AMP_AVDEC_HDLR_s** is defined in `Decode.h`) Please refer to Section 7.2.4.1 for more details.
AMP_YUV_BUFFER_S	**buf**	Buffer to display. Size of AOI must be the same as the Vout size. (**AMP_YUV_BUFFER_s** is defined in `common.h`) Please refer to Section 7.4.7.1.1 for more details

Table 7-67. Parameters for Codec SDK6 API **AmpVideoDec_DumpFrame()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-68. Returns for Codec SDK6 API **AmpVideoDec_DumpFrame()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.5.7 AmpVideoDec_GetTime

API Syntax:

AmpVideoDec_GetTime (AMP_AVDEC_HDLR_s * Hdlr, UINT64 * Time)

Function Description:

- This function is used to get the current time.

Parameters:

Type	Parameter	Description
AMP_AVDEC_HDLR_s	**hdlr**	Codec handler structure for video decode (**AMP_AVDEC_HDLR_s** is defined in `Decode.h`) Please refer to Section 7.2.4.1 for more details.
UINT64	**Time**	Current time in ms

Table 7-69. Parameters for Codec SDK6 API **AmpVideoDec_GetTime()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-70. Returns for Codec SDK6 API **AmpVideoDec_GetTime()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.5.8 AmpVideoDec_SetEosPts

API Syntax:

AmpVideoDec_SetEosPts (AMP_AVDEC_HDLR_s * Hdlr, UINT64 EosFileTime, UINT32 DeltaFileTime, UINT32 FileTimeScale)

Function Description:

- This function is used to set EOS points.

Parameters:

Type	Parameter	Description
AMP_AVDEC_HDLR_s	**hdlr**	Codec handler structure for video decode (**AMP_AVDEC_HDLR_s** is defined in `Decode.h`) Please refer to Section 7.2.4.1 for more details.
UINT64	**EosFileTime**	EOS time (of file)
UINT32	**DeltaFileTime**	EOS delta. It's time (of file) per frame
UINT32	**FileTimeScale**	Time (of file) per second

Table 7-71. Parameters for Codec SDK6 API **AmpVideoDec_SetEosPts()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-72. Returns for Codec SDK6 API **AmpVideoDec_SetEosPts()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.5.9 AmpVideoDec_GetXCodec

API Syntax:

AmpVideoDec_GetXCodec (AMP_AVDEC_HDLR_s * hdlr, AMP_AVDEC_HDLR_s ** xcodeHdlr)

Function Description:

- This function is used to get the transcode codec.

Parameters:

Type	Parameter	Description
AMP_AVDEC_HDLR_s	**hdlr**	Codec handler structure for video decode (**AMP_AVDEC_HDLR_s** is defined in `Decode.h`) Please refer to Section 7.2.4.1 for more details.
AMP_AVDEC_HDLR_s	**xcodeHdlr**	Transcode codec (**AMP_AVDEC_HDLR_s** is defined in `Decode.h`) Please refer to Section 7.2.4.1 for more details.

Table 7-73. Parameters for Codec SDK6 API **AmpVideoDec_GetXCodec()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-74. Returns for Codec SDK6 API **AmpVideoDec_GetXCodec()**

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.5.10 AmpVideoDec_SetFadingEffect

API Syntax:

AmpVideoDec_SetFadingEffect (AMP_AVDEC_HDLR_s * hdlr, UINT8 numFadingCfg, AMP_VIDEODEC_FADING_EFFECT_s * fadingCfg)

Function Description:

- This function is used to set the fading effect on video playback.

Parameters:

Type	Parameter	Description
AMP_AVDEC_HDLR_s	**hdlr**	Codec handler structure for video decode (**AMP_AVDEC_HDLR_s** is defined in `Decode.h`) Please refer to Section 7.2.4.1 for more details.
UINT8	**numFadingCfg**	Number of fading effect. Maximum value is 2 to support fade in and fade out.
AMP_VIDEO-DEC_FADING_EFFECT_s	**fadingCfg**	

Table 7-75. Parameters for Codec SDK6 API **AmpVideoDec_SetFadingEffect()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-76. Returns for Codec SDK6 API **AmpVideoDec_SetFadingEffect()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

.

7.5.10.1 AmpVideoDec_SetFadingEffect> AMP_VIDEODEC_FADING_EFFECT_s

Type	Field	Description
UINT32	**StartTime**	Fading start time in ms
UINT32	**Duration**	Fading duration in ms, 0 to disable
UINT16	**StartMatrix**	Fading start matrix
INT16	**StartYOffset**	
INT16	**StartUOffset**	

Type	Field	Description
INT16	**StartVOffset**	
UINT16	**EndMatrix**	Fading end matrix
INT16	**EndYOffset**	
INT16	**EndUOffset**	
INT16	**EndVOffset**	

*Table 7-77. Definition of **AMP_VIDEODEC_FADING_EFFECT_s** for Codec SDK6 API **AmpVideoDec_SetFadingEffect**().*

7.6 Codec: List of APIs for AudioEnc

This section lists the Amba audio encoder APIs. The implementation of the Amba audio encode module includes the initialization Amba audio encode module, create Amba audio encode module, delete Amba audio encode module, and get Amba audio encode module functions..

- AmpAudioEnc_GetInitDefaultCfg
- AmpAudioEnc_Init
- AmpAudioEnc_GetDefaultCfg
- AmpAudioEnc_Create
- AmpAudioEnc_Delete
- AmpAudioEnc_Config

7.6.1 AmpAudioEnc_GetInitDefaultCfg

API Syntax:

AmpAudioEnc_GetInitDefaultCfg (AMP_AUDIOENC_INIT_CFG_s * defInitCfg)

Function Description:

- This function is used to get audio encode module default configuration for initialization.

Parameters:

Type	Parameter	Description
AMP_AUDIOENC_INIT_CFG_s	**defInitCfg**	Initial configuration

Table 7-78. Parameters for Codec SDK6 API **AmpAudioEnc_GetInitDefaultCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-79. Returns for Codec SDK6 API **AmpAudioEnc_GetInitDefaultCfg()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.6.2 AmpAudioEnc_Init

API Syntax:

AmpAudioEnc_Init (AMP_AUDIOENC_INIT_CFG_s * cfg)

Function Description:

- This function is used to initialize audio encode module. The function should only be invoked once. The user MUST invoke this function before using audio encode module. The memory pool of the module will be provided by the user.

Parameters:

Type	Parameter	Description
AMP_AUDIO-ENC_INIT_CFG_s	**cfg**	Audio Encode module configuration

Table 7-80. Parameters for Codec SDK6 API **AmpAudioEnc_Init()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-81. Returns for Codec SDK6 API **AmpAudioEnc_Init()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.6.3 AmpAudioEnc_GetDefaultCfg

API Syntax:

AmpAudioEnc_GetDefaultCfg (AMP_AUDIOENC_HDLR_CFG_s * defCfg)

Function Description:

- This function is used to get the full default setting of the audio codec.

Parameters:

Type	Parameter	Description
AMP_AUDIO-ENC_HDLR_CFG_s	**defCfg**	Default Configuration

Table 7-82. Parameters for Codec SDK6 API **AmpAudioEnc_GetDefaultCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-83. Returns for Codec SDK6 API **AmpAudioEnc_GetDefaultCfg()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.6.3.1 AmpAudioEnc_GetDefaultCfg> AMP_AUDIOENC_HDLR_CFG_s

Type	Field	Description
AMBA_AUDIO_TYPE_e	**EncType**	Audio Encode Type
UINT8	**EventDataReadySkipNum**	Fire “data ready” event to FIFO per (1 + evenDataReady-SkipNum) frames
UINT8	**EncoderTaskPriority**	Audio encoder task priority
UINT8	**InputTaskPriority**	Audio input receiver task priority
UINT32	**FadeInTime**	Fade in when encode starts
UINT32	**FadeOutTime**	Fade out when encode starts
UINT32	**SrcSampleRate**	Sample frequency of the input
UINT32	**DstSampleRate**	Sample frequency of the output

Type	Field	Description
UINT16	**SrcChannelMode**	Channel mode of the input
UINT16	**DstChannelMode**	Channel mode of the output
UINT8	**NumInstance**	Number of audio instance in this handler
AMP_AUDIO_INSTANCE_s*	**AudioInstance**	Audio encode instance relative to audio_lib
union	**Spec**	Audio encode specification
AMP_CALLBACK_f	**cbCfgUpdated**	Setting take effect
AMP_CALLBACK_f	**cbEvent**	Event related to the encoder
AMP_ENC_BITSBUFFER_CFG_s	**BitsBufCfg**	Buffer configuration
INT8*	**CalibCurveAddr**	Calib Curve Address
INT32*	**CalibworkBuffer**	Calibration working buffer address

*Table 7-84. Definition of **AMP_AUDIOENC_HDLR_CFG_s** for Codec SDK6 API **AmpAudioEnc_GetDefaultCfg**().*

7.6.4 AmpAudioEnc_Create

API Syntax:

AmpAudioEnc_Create (AMP_AUDIOENC_HDLR_CFG_s * cfg, AMP_AVENC_HDLR_s ** aencHdlr)

Function Description:

- This function is used to create an audio handler and set the initial configuration.

Parameters:

Type	Parameter	Description
AMP_AUDIOENC_HDLR_CFG_s	**cfg**	AudioEnc handler configuration
AMP_AVENC_HDLR_s	**aencHdlr**	Common handler structure for video encode

Table 7-85. Parameters for Codec SDK6 API **AmpAudioEnc_Create()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-86. Returns for Codec SDK6 API **AmpAudioEnc_Create()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.6.5 AmpAudioEnc_Delete

API Syntax:

AmpAudioEnc_Delete (AMP_AVENC_HDLR_s * hdlr)

Function Description:

- This function is used to delete an audio handler.

Parameters:

Type	Parameter	Description
AMP_AVENC_ HDLR_s	**hdlr**	Audio encode instance

Table 7-87. Parameters for Codec SDK6 API **AmpAudioEnc_Delete()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-88. Returns for Codec SDK6 API **AmpAudioEnc_Delete()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.6.6 AmpAudioEnc_Config

API Syntax:

AmpAudioEnc_Config (AMP_AVENC_HDLR_s * hdlr, AMP_AUDIOENC_CFG_s * cfg)

Function Description:

- This function is used to configure the audio encode handler.

Parameters:

Type	Parameter	Description
AMP_AVENC_ HDLR_s	**hdlr**	Audio encode instance
AMP_AUDIO-ENC_CFG_s	**cfg**	Encoder configuration

Table 7-89. Parameters for Codec SDK6 API **AmpAudioEnc_Config()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-90. Returns for Codec SDK6 API **AmpAudioEnc_Config()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.7 Codec: List of APIs for DummyEnc

This section lists the sample encoder APIs. The implementation of the sample encode module includes create sample encode module and delete sample encode module functions..

- AmpDummyEnc_GetDefaultCfg
- AmpDummyEnc_Create
- AmpDummyEnc_Delete

7.7.1 AmpDummyEnc_GetDefaultCfg

API Syntax:

AmpDummyEnc_GetDefaultCfg (AMP_DUMMYENC_CFG_s * cfg)

Function Description:

- This function is used to get full default settings of the module.

Parameters:

Type	Parameter	Description
AMP_DUMMY-ENC_CFG_s	**cfg**	Encode video configuration

Table 7-91. Parameters for Codec SDK6 API **AmpDummyEnc_GetDefaultCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-92. Returns for Codec SDK6 API **AmpDummyEnc_GetDefaultCfg()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.7.1.1 AmpDummyEnc_GetDefaultCfg> AMP_DUMMY_ENC_CFG_s

Type	Field	Description
AMBA_YUV_COLOR_s	**bgColor**	Background color
AMP_CALLBACK_f	**cbCfgUpdated**	Callback when configuration is updated
AMP_CALLBACK_f	**cbCodecEvent**	Callback event
WCHAR	**rawFn[DUMMY_MAX_FILENAME_LENGTH]**	Raw data file name
WCHAR	**idxFn[DUMMY_MAX_FILENAME_LENGTH]**	Index data file name
UINT8	**idrItvl**	IDR interval

Type	Field	Description
Char *	**rawBuffer**	Raw data buffer address
UINT32	**rawBufferSize**	Raw data buffer size
Char *	**descBuffer**	Descriptor buffer
UINT32	**numDescBuffer**	Descriptor buffer entry
UINT32	**frmDly**	Frame sent delay
UINT32	**memRunOutAlarmThr**	runout memory threshold

*Table 7-93. Definition of **AMP_DUMMY_ENC_CFG_s** for Codec SDK6 API **AmpDummyEnc_GetDefaultCfg**().*

7.7.2 AmpDummyEnc_Create

API Syntax:

AmpDummyEnc_Create (AMP_DUMMYENC_CFG_s * cfg, AMP_AVENC_HDLR_s ** hdlr)

Function Description:

- This function is used to set the initial configuration for the video encode module.

Parameters:

Type	Parameter	Description
AMP_DUMMY-ENC_CFG_s	**cfg**	Encode video configuration
AMP_AVENC_HDLR_s	**hdlr**	Common handler structure for video encode

Table 7-94. Parameters for Codec SDK6 API **AmpDummyEnc_Create()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-95. Returns for Codec SDK6 API **AmpDummyEnc_Create()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.7.3 AmpDummyEnc_Delete

API Syntax:

AmpDummyEnc_Delete (AMP_AVENC_HDLR_s * hdlr)

Function Description:

- This function is used to delete the configuration for the video encode module

Parameters:

Type	Parameter	Description
AMP_AVENC_ HDLR_s	**hdlr**	Common handler structure for video encode

Table 7-96. Parameters for Codec SDK6 API **AmpDummyEnc_Delete()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-97. Returns for Codec SDK6 API **AmpDummyEnc_Delete()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.8 Codec: List of APIs for StillEnc

This section lists the still encoder APIs. The implementation of the still encode module includes initial still encode module, create still encode module, delete still encode module, config still encode module window layout, trigger to start raw capture, update window layout in IDLE state, control LiveView capture between LiveView and Still-Capture functions..

- AmpStillEnc_GetInitDefaultCfg
- AmpStillEnc_Init
- AmpStillEnc_GetDefaultCfg
- AmpStillEnc_Create
- AmpStillEnc_Delete
- AmpStillEnc_ConfigVinMain
- AmpStillEnc_StartRawCapture
- AmpStillEnc_StartFollowingRawCapture
- AmpStillEnc_UpdateVinMain
- AmpStillEnc_EnableLiveviewCapture

7.8.1 AmpStillEnc_GetInitDefaultCfg

API Syntax:

AmpStillEnc_GetInitDefaultCfg (AMP_STILLENC_INIT_CFG_s * defInitCfg)

Function Description:

- This function is used to get the still encoder module default configuration for initialization.

Parameters:

Type	Parameter	Description
AMP_ STILLENC_INIT_ CFG_s	**defInitcfg**	Initial default configuration

Table 7-98. Parameters for Codec SDK6 API **AmpStillEnc_GetInitDefaultCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-99. Returns for Codec SDK6 API **AmpStillEnc_GetInitDefaultCfg()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.8.1.1 AmpStillEnc_GetInitDefaultCfg> AMP_STILLENC_INIT_CFG_s

Type	Field	Description
UINT8	**MaxLayerPerEncoder**	Maximum number of layers per encoder layout
AMP_YUV_ COLOR_s	**BgColor**	Background color of LiveView
AMP_TASK_ INFO_s	**MainTaskInfo**	General task information
AMP_TASK_ INFO_s	**RawCapTaskInfo**	Raw capture task information
AMP_TASK_ INFO_s	**RawEncTaskInfo**	Raw encode task information
UINT32	**MsgQueueNum**	Number of entries per message queue
UINT8*	**MemoryPoolAddr**	Working buffer start address

Type	Field	Description
UINT32	**MemoryPoolSize**	Size of the buffer
UINT32	**ScrptBufSize**	Size of Still encode script buffer

*Table 7-100. Definition of **AMP_STILLENC_INIT_CFG_s** for Codec SDK6 API **AmpStillEnc_GetInitDefaultCfg**().*

7.8.2 AmpStillEnc_Init

API Syntax:

AmpStillEnc_Init (AMP_STILLENC_INIT_CFG_s * cfg)

Function Description:

- This function is used to initialize the still encode module. The function should only be invoked once. The user must invoke this function before using the still encode module. The memory pool of the module will be provided by the user.

Parameters:

Type	Parameter	Description
AMP_ STILLENC_INIT_ CFG_s	**defInitcfg**	Initial default configuration

Table 7-101. Parameters for Codec SDK6 API **AmpStillEnc_Init()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-102. Returns for Codec SDK6 API **AmpStillEnc_Init()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.8.3 AmpStillEnc_GetDefaultCfg

API Syntax:

AmpStillEnc_GetDefaultCfg (AMP_STILLENC_HDLR_CFG_s * defCfg)

Function Description:

- This function is sed to get the full default setting of the module.

Parameters:

Type	Parameter	Description
AMP_ STILLENC_ HDLR_CFG_s	**defCfg**	Still encoder handler configuration. Please see Section 7.8.3.1 for more details.

Table 7-103. Parameters for Codec SDK6 API **AmpStillEnc_Init().**

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-104. Returns for Codec SDK6 API **AmpStillEnc_Init().**

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.8.3.1 AmpStillEnc_GetDefaultCfg> AMP_STILLENC_HDLR_CFG_s

Type	Field	Description
AMP_SYSTEM_ FREQ_s	**SysFreq**	System frequency when handler state is not idle. Primary stream handler only.
UINT8*	**DspWorkBufAddr**	DSP working area address, only AMP_VIDEOENC_STREAM_PRIMARY check this field, 0 = no change
UINT32	**DspWorkBufSize**	DSP working area size, only AMP_VIDEOENC_STREAM_PRIMARY check this field, 0 = no change
AMP_ CALLBACK_f	**cbCfgUpdated**	Setting take effect
AMP_ CALLBACK_f	**cbEvent**	Event related to the encoder
UINT8:2	**EnableMultiView**	Enable MultiView or not

Type	Field	Description
UINT8:1	**LiveViewProcMode**	Enable Hybrid LiveView or not
UINT8:2	**LiveViewAlgoMode**	Algo Mode in Hybrid proc mode
UINT8:1	**LiveViewOSMode**	Enable express oversampling or not
UINT8:1	**LiveViewHdrMode**	Enable Hybrid HDR or not, A12 only
UINT8:1	**Interlace**	Interlaced encoding
AMP_VIDEOENC_LAYOUT_CFG_s	**MainLayout**	Main Window layout
UINT32	**MainTimeScale**	Frame rate time scale
UINT32	**MainTickPerPicture**	Frame rate tick per picture. Frame_rate = TimeScale/TickPerPicture/(1 + interlace) 29.97i: Timescale = 60000, TickPerPic = 1001: frame_rate = 60000/1001/(1+1)=29.97 59.94p: Timescale = 60000, TickPerPic = 1001: frame_rate = 60000/1001/(1+0)=59.94
AMP_ENC_BITSBUFFER_CFG_s	**BitsBufCfg**	Buffer config
AMP_VIDEOENC_BITSTREAM_CFG_s	**BitsCfg**	Bitstream config
UINT8	**LiveviewOBModeEnable**	Enable Liveview OB (Optical Black)

*Table 7-105. Definition of **AMP_STILLENC_HDLR_CFG_s** for Codec SDK6 API **AmpStillEnc_GetDefaultCfg**().*

7.8.4 AmpStillEnc_Create

API Syntax:

AmpStillEnc_Create (AMP_STILLENC_HDLR_CFG_s * cfg, AMP_STLENC_HDLR_s ** sencHdlr)

Function Description:

- This function is used to create an still encoder pipeline and set the initial configuration.

Parameters:

Type	Parameter	Description
AMP_ STILLENC_ HDLR_CFG_s	**cfg**	Still encoder handler configuration
AMP_STLENC_ HDLR_s	**sencHdlr**	Common handler structure for still encode. Please see Section 7.8.4.1 for more details.

Table 7-106. Parameters for Codec SDK6 API **AmpStillEnc_Create()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-107. Returns for Codec SDK6 API **AmpStillEnc_Create()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.8.4.1 AmpStillEnc_Create> AMP_STLENC_HDLR_s

Type	Field	Description
Void*	**ctx**	Pointer to CODEC context
	function	Pointer to CODEC operation functions

*Table 7-108. Definition of **AMP_STLENC_HDLR_s** for Codec SDK6 API **AmpStillEnc_Create**().*

7.8.5 AmpStillEnc_Delete

API Syntax:

AmpStillEnc_Delete (AMP_STLENC_HDLR_s * hdlr)

Function Description:

- This function deletes a still encoder pipeline.

Parameters:

Type	Parameter	Description
AMP_STLENC_HDLR_s	**sencHdlr**	Common handler structure for still encode. Please see Section 7.8.4.1 for more details.

Table 7-109. Parameters for Codec SDK6 API **AmpStillEnc_Delete()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-110. Returns for Codec SDK6 API **AmpStillEnc_Delete()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.8.6 AmpStillEnc_ConfigVinMain

API Syntax:

AmpStillEnc_ConfigVinMain (UINT32 numVin, AMP_VIN_RUNTIME_CFG_s * vinCfg, AMP_STILLENC_MAIN_CFG_s * mainCfg)

Function Description:

- This function is the template for getting the configuration.

Parameters:

Type	Parameter	Description
UINT32	**numVin**	Number of VIN to be configured
AMP_VIN_RUNTIME_CFG_s	**vinCfg**	VIN configuration array. Please see Section 7.8.6.1 for more details.
AMP_STILLENC_MAIN_CFG_s	**mainCfg**	Main window configuration array. Please see Section 7.8.6.2 for more details.

Table 7-111. Parameters for Codec SDK6 API **AmpStillEnc_ConfigVinMain()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-112. Returns for Codec SDK6 API **AmpStillEnc_ConfigVinMain()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.8.6.1 AmpStillEnc_ConfigVinMain> AMP_VIN_RUNTIME_CFG_s

Type	Field	Description
AMBA_VIN_HDLR_s*	**Hdlr**	Point to VIN handler
AMP_AREA_s	**HwCaptureWindow**	HW capture window
AMBA_SENSOR_MODE_ID_u	**Mode**	Mode ID
UINT32	**LayourNumber**	How many main windows will be produced from this VIN

Type	Field	Description
AMP_VIN_LAYOUT_CFG_s*	**Layout**	Layout list

Table 7-113. Definition of ***AMP_VIN_RUNTIME_CFG_s*** *for Codec SDK6 API* ***AmpStillEnc_ConfigVinMain****().*

7.8.6.2 AmpStillEnc_ConfigVinMain> AMP_STILLENC_MAIN_CFG_s

Type	Field	Description
AMBA_STLENC_HDLR_s*	**Hdlr**	Point to Still encode handler
AMP_SYSTEM_FREQ_s	**SysFreq**	System frequency when handler state is not idle. Primary stream handler only.
UINT8*	**DspWorkBufAddr**	DSP working area address, only AMP_VIDEOENC_STREAM_PRIMARY checks this field, 0 = no change
UINT32	**DspWorkBufSize**	DSP working area size, only AMP_VIDEOENC_STREAM_PRIMARY check this field, 0 = no change
UINT8:1	**LiveViewProcMode**	Enable Hybrid LiveView or not
UINT8:4	**LiveViewAlgoMode**	Algo Mode in Hybrid proc mode
UINT8:1	**LiveViewOSMode**	Enable Express OverSampling or not
UINT8:1	**LiveViewHdrMode**	Enable Hybrid HDR or not, A12 only
UINT8:1	**Interlace**	Interlaced encoding
AMP_VIDEOENC_LAYOUT_CFG_s	**MainLayout**	Main window layout
UINT32	**MainTimeScale**	Frame rate time scale
UINT32	**MainTickPerPicture**	Frame rate tick per picture. Frame_rate = TimeScale/TickPerPicture/(1 + interlace) 29.97i: Timescale = 6000, TickPerPic = 1001: frame_rate = 6000/1001/(1+1)=29.97 29.97p: timescale = 3000, TickPerPic = 1001: frame_rate = 3000/1001/(1+0)=29.97 59.94p: Timescale = 6000, TickPerPic = 1001: frame_rate = 6000/1001/(1+0)=59.94

Table 7-114. Definition of ***AMP_STILLENC_MAIN_CFG_s*** *for Codec SDK6 API* ***AmpStillEnc_ConfigVinMain****().*

7.8.7 AmpStillEnc_StartRawCapture

API Syntax:

AmpStillEnc_StartRawCapture (AMP_STLENC_HDLR_s * hdlr, AMP_STILLENC_RAWCAP_DSP_CTRL_s * ctrl)

Function Description:

- This function is used to start raw capture. This API should be invoked after LiveView stops and the sensor program is completed.

Parameters:

Type	Parameter	Description
AMP_STLENC_HDLR_s	**sencHdlr**	Common handler structure for still encode
AMP_STILLENC_RAWCAP_DSP_CTRL_s	**ctrl**	DSP control information. Please see Section 7.8.7.1 for more details.

Table 7-115. Parameters for Codec SDK6 API **AmpStillEnc_StartRawCapture()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-116. Returns for Codec SDK6 API **AmpStillEnc_StartRawCapture()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.8.7.1 AmpStillEnc_StartRawCapture> AMP_STILLENC_RAWCAP_DSP_CTRL_s

Type	Field	Description
AMBA_DSP_CHANNEL_ID_u	**VinChan**	VIN channel
UINT8	**VidSkip**	Skip frame number
UINT32	**RawCapNum**	Raw capture number in one round

Type	Field	Description
UINT32	**StillProc**	Still Process mode

*Table 7-117. Definition of **AMP_STILLENC_RAWCAP_DSP_CTRL_s** for Codec SDK6 API **AmpStillEnc_StartRawCapture**().*

7.8.8 AmpStillEnc_StartFollowingRawCapture

API Syntax:

AmpStillEnc_StartFollowingRawCapture (AMP_STLENC_HDLR_s * hdlr, AMP_STILLENC_RAWCAP_DSP_CTRL_s * dspCtrl)

Function Description:

- This function is used to start following raw capture. This API should be invoked after the first raw capture is done.

Parameters:

Type	Parameter	Description
AMP_STLENC_ HDLR_s	**sencHdlr**	Common handler structure for still encode
AMP_ STILLENC_ RAWCAP_DSP_ CTRL_s	**ctrl**	DSP control information. Please see Section 7.8.7.1 for more details.

*Table 7-118. Parameters for Codec SDK6 API **AmpStillEnc_StartFollowingRawCapture()**.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 7-119. Returns for Codec SDK6 API **AmpStillEnc_StartFollowingRawCapture()**.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.8.9 AmpStillEnc_UpdateVinMain

API Syntax:

AmpStillEnc_UpdateVinMain (UINT32 numVin, AMP_VIN_RUNTIME_CFG_s * vinCfg, AMP_STILLENC_MAIN_CFG_s * mainCfg)

Function Description:

- This function is used to update VIN/MAIN configuration when in IDLE state.

Parameters:

Type	Parameter	Description
UINT32	**numVin**	Number of VIN to be configured
AMP_VIN_RUNTIME_CFG_s	**vinCfg**	VIN configuration array
AMP_STILLENC_MAIN_CFG_s	**mainCfg**	Main window configuration array. Please see Section 7.8.6.2 for more details.

Table 7-120. Parameters for Codec SDK6 API ***AmpStillEnc_UpdateVinMain()***.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-121. Returns for Codec SDK6 API ***AmpStillEnc_UpdateVinMain()***.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.8.10 AmpStillEnc_EnableLiveviewCapture

API Syntax:

AmpStillEnc_EnableLiveviewCapture (UINT8 Enable, UINT8   ShowVideoPlane)

Function Description:

- This function is used to control LiveView capture between LiveView and StillCapture.

Parameters:

Type	Parameter	Description
UINT8	**Enable**	Enable or disable liveview capture: AMP_STILL_STOP_LIVEVIEW_0 AMP_STILL_PREPARE_TO_VIDEO_LIVEVIEW_1 AMP_STILL_PREPARE_TO_STILL_CAPTURE_2 AMP_STILL_PREPARE_TO_STILL_LIVEVIEW_3 AMP_STILL_SWITCH_TO_STILL_LIVEVIEW_4
UINT8	**ShowVideoPlane**	Show Video Plane

Table 7-122. Parameters for Codec SDK6 API **AmpStillEnc_EnableLiveviewCapture()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-123. Returns for Codec SDK6 API **AmpStillEnc_EnableLiveviewCapture()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9 Codec: List of APIs for VideoEnc

This section lists the video encoder APIs. The implementation of the video encode module includes initial video encode module, create video encode module, delete video encode module, config video encode module window layout, config bitrate/quality control, config Bitstream buffer, trigger TimeLapse video capture, config encode blending, and trigger VideoTuning flow functions..

- AmpVideoEnc_GetInitDefaultCfg
- AmpVideoEnc_Init
- AmpVideoEnc_GetDefaultCfg
- AmpVideoEnc_Create
- AmpVideoEnc_Delete
- AmpVideoEnc_ConfigVinMain
- AmpVideoEnc_GetLayout
- AmpVideoEnc_SetBitstreamConfig
- AmpVideoEnc_SetBitstreamBuffer
- AmpVideoEnc_GetBitstreamConfig
- AmpVideoEnc_GetDefaultH264Header
- AmpVideoEnc_SetRuntimeBitrate
- AmpVideoEnc_SetRuntimeQuality
- AmpVideoEnc_CaptureTimeLapsedFrame
- AmpVideoEnc_SetRuntimeFrameRate
- AmpVideoEnc_SetEncodeBlend
- AmpVideoEnc_FeedEncodeRaw
- AmpVideoEnc_GetEncodingInfo

7.9.1 AmpVideoEnc_GetInitDefaultCfg

API Syntax:

AmpVideoEnc_GetInitDefaultCfg (AMP_VIDEOENC_INIT_CFG_s * defInitCfg)

Function Description:

- This function is used to get video encoder module default configuration for initialization.

Parameters:

Type	Parameter	Description
AMP_VIDEO-ENC_INIT_CFG_s	**defInitCfg**	Video Encode Module Configuration. Please refer to Section 7.9.1.1 for more details.

Table 7-124. Parameters for Codec SDK6 API **AmpVideoEnc_GetInitDefaultCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-125. Returns for Codec SDK6 API **AmpVideoEnc_GetInitDefaultCfg()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.1.1 AmpVideoEnc_GetInitDefaultCfg> AMP_VIDEOENC_INIT_CFG_s

Type	Field	Description
UINT8	**MaxLayerPerEncoder**	Maximum number of layers per encoder layout
AMP_YUV_COLOR_s	**BackgroundColor**	Background color of video
AMP_TASK_INFO_s	**TaskInfo**	vdsp/general task information
UINT32	**MsgQueueNumber**	Number of entries per message queue
UINT8*	**MemoryPoolAddr**	Working buffer start address
UINT32	**MemoryPoolSize**	Sixe of the buffer

Table 7-126. Definition of ***AMP_VIDEOENC_INIT_CFG_s*** *for Codec SDK6 API* ***AmpVideoEnc_GetInitDefaultCfg****().*

7.9.2 AmpVideoEnc_Init

API Syntax:

AmpVideoEnc_Init (AMP_VIDEOENC_INIT_CFG_s * cfg)

Function Description:

- This function is used to initialize the video encode module. This function should be invoked only once. The user must invoke this function before using the video encode module. The memory pool of the module will be provided by the user.

Parameters:

Type	Parameter	Description
AMP_VIDEO-ENC_INIT_CFG_s	**cfg**	Video Encode module configuration. Please refer to Section 7.9.1.1 for more details.

Table 7-127. Parameters for Codec SDK6 API **AmpVideoEnc_Init()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-128. Returns for Codec SDK6 API **AmpVideoEnc_Init()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.3 AmpVideoEnc_GetDefaultCfg

API Syntax:

AmpVideoEnc_GetDefaultCfg (AMP_VIDEOENC_HDLR_CFG_s * defCfg)

Function Description:

- This function is used to get the full default setting of the module.

Parameters:

Type	Parameter	Description
AMP_VIDEO-ENC_HDLR_CFG_s	**defCfg**	Configuration. Please see Section 7.9.3.1 for more details.

Table 7-129. Parameters for Codec SDK6 API **AmpVideoEnc_GetDefaultCfg()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-130. Returns for Codec SDK6 API **AmpVideoEnc_GetDefaultCfg()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.3.1 AmpVideoEnc_GetDefaultCfg> AMP_VIDEOENC_HDLR_CFG_s

Type	Field	Description
AMP_VIDEO-ENC_STREAM_ID_e	**StreamId**	Stream ID
UINT8	**EventDataReadySkipNum**	Fire “data ready” event to FIFO per (1 + evenDataReadySkipNum) frames
AMP_SYSTEM_FREQ_s	**SysFreq**	System frequency when handler state is not idle. Primary stream handler only
UINT8*	**DspWorkBufAddr**	DSP working area address, only AMP_VIDEOENC_STREAM_PRIMARY checks this field, 0 = no change
UINT32	**DspWorkBufSize**	DSP working area size, only AMP_VIDEOENC_STREAM_PRIMARY checks this field, 0 = no change

Type	Field	Description
AMP_CALLBACK_f	**cbCfgUpdated**	Setting take effect
AMP_CALLBACK_f	**cbEvent**	Event related to the encoder
UINT8:1	**EnableLowDelayLiveview**	Enable low delay mode or not. A9 only. Other chips no effect
UINT8:2	**EnableMultiView**	Enable MultiView or not
UINT8:1	**LiveViewProcMode**	Enable Hybrid LiveView or not. A12 only.
UINT8:2	**LiveViewAlgoMode**	Algo mode in hybrid proc mode. A12 only.
UINT8:1	**LiveViewOSMode**	Enable Express OverSampling or not. A12 only.
UINT8:1	**LiveViewHdrMode**	Enable Hybrid HDR or not, A12 only
UINT8:1	**Interlace**	Interlaced encoding
AMP_VIDEOENC_LAYOUT_CFG_s	**MainLayout**	Main window layout
UINT32	**MainTimeScale**	Frame rate time scale
UINT32	**MainTickPerPicture**	Frame rate tick per picture. Frame_rate = TimeScale/TickPerPicture/(1+interlace) 29.97i: Timescale = 6000, TickPerPic = 1001: frame_rate = 6000/1001/(1+1)=29.97 59.94p: Timescale = 6000, TickPerPic = 1001: frame_rate = 6000/1001/(1+0)=59.94
AMP_ENC_BITSBUFFER_CFG_s	**BitsBufCfg**	Buffer configuration
AMP_VIDEOENC_BITSTREAM_CFG_s	**BitsCfg**	Bitstream configuration
AMP_VIN_SOURCE_e	**VinSource**	Vin source
UINT8	**LiveviewOBModeEnable**	Enable Liveview OB (Optical Black)

*Table 7-131. Definition of **AMP_VIDEOENC_HDLR_CFG_s** for Codec SDK6 API **AmpVideoEnc_GetDefaultCfg**().*

7.9.4 AmpVideoEnc_Create

API Syntax:

AmpVideoEnc_Create (AMP_VIDEOENC_HDLR_CFG_s * cfg, AMP_AVENC_HDLR_s ** vencHdlr)

Function Description:

- This function is used to set the initial configuration for the video encode module.

Parameters:

Type	Parameter	Description
AMP_VIDEO-ENC_HDLR_CFG_s	**Cfg**	Configuration. Please see Section 7.9.3.1 for more details.
AMP_AVENC_HDLR_s	**vencHdlr**	Encoder Instance pointer. Please see Section 7.9.4.1 for more details.

Table 7-132. Parameters for Codec SDK6 API **AmpVideoEnc_Create()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-133. Returns for Codec SDK6 API **AmpVideoEnc_Create()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.4.1 AmpVideoEnc_Create> AMP_AVENC_HDLR_s

Type	Field	Description
Void*	**ctx**	Pointer to CODEC context
AMP_AVENC_CODEC_s*	**function**	Pointer to CODEC operation functions

Table 7-134. Definition of ***AMP_AVENC_HDLR_s*** *for Codec SDK6 API* ***AmpVideoEnc_Create****().*

7.9.5 AmpVideoEnc_Delete

API Syntax:

AmpVideoEnc_Delete (AMP_AVENC_HDLR_s * hdlr)

Function Description:

- This function is used to delete instance.

Parameters:

Type	Parameter	Description
AMP_AVENC_ HDLR_s	**hdlr**	Encoder instance. Please see Section 7.9.4.1 for more details.

Table 7-135. Parameters for Codec SDK6 API **AmpVideoEnc_Delete()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-136. Returns for Codec SDK6 API **AmpVideoEnc_Delete()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.6 AmpVideoEnc_ConfigVinMain

API Syntax:

AmpVideoEnc_ConfigVinMain (UINT32 numVin, AMP_VIN_RUNTIME_CFG_s * vinCfg, UINT32 numMain, AMP_VIDEOENC_MAIN_CFG_s * mainCfg)

Function Description:

- This function is used to configure VIN/MAIN after the handler is created.

Parameters:

Type	Parameter	Description
UINT32	**numVin**	Number of VINs to be configured
AMP_VIN_RUNTIME_CFG_s	**vinCfg**	VIN configuration array
UINT32	**numMain**	Number of main windows to be configured
AMP_VIDEOENC_MAIN_CFG_s	**mainCfg**	Main window configuration array. Please refer to Section 7.9.6.1 for more details.

Table 7-137. Parameters for Codec SDK6 API **AmpVideoEnc_ConfigVinMain()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-138. Returns for Codec SDK6 API **AmpVideoEnc_ConfigVinMain()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.6.1 AmpVideoEnc_ConfigVinMain> AMP_VIDEOENC_MAIN_CFG_s

Type	Field	Description
AMP_AVENC_HDLR_s*	**Hdlr**	Point to which VideoEncode handler
UINT16	**EventDataReadySkipNum**	Fire “data ready” event to FIFO per (1 + evenDataReadySkipNum) frames
AMP_SYSTEM_FREQ_s	**SysFreq**	System frequency when handler state is not idle. Primary stream handler only.

Type	Field	Description
UINT8*	**DspWorkBufAddr**	DSP working area address, only AMP_VIDEOENC_STREAM_PRIMARY checks this field, 0 = no change
UINT32	**DspWorkBufSize**	DSP working area size, only AMP_VIDEOENC_STREAM_PRIMARY check this field, 0 = no change
UINT8:1	**LiveViewProcMode**	Enable Hybrid LiveView or not, A12 only
UINT8:4	**LiveViewAlgoMode**	Algo Mode in Hybrid proc mode, A12 only
UINT8:1	**LiveViewOSMode**	Enable Express OverSampling or not, A12 only
UINT8:1	**LiveViewHdrMode**	Enable Hybrid HDR or not, A12 only
UINT8:1	**Interlace**	Interlaced encoding
AMP_VIDEOENC_LAYOUT_CFG_s	**MainLayout**	Main window layout
UINT32	**MainTimeScale**	Frame rate time scale
UINT32	**MainTickPerPicture**	Frame rate tick per picture. Frame_rate = TimeScale/TickPerPicture/(1+interlace) 29.97i: Timescale = 6000, TickPerPic = 1001: frame_rate = 6000/1001/(1+1)=29.97 59.94p: Timescale = 6000, TickPerPic = 1001: frame_rate = 6000/1001/(1+0)=59.94

*Table 7-139. Definition of **AMP_VIDEOENC_MAIN_CFG_s** for Codec SDK6 API **AmpVideoEnc_Create**().*

7.9.7 AmpVideoEnc_GetLayout

API Syntax:

AmpVideoEnc_GetLayout (AMP_AVENC_HDLR_s * hdlr, AMP_VIDEOENC_LAYOUT_CFG_s * layout)

Function Description:

- This function is used to get the current layout.

Parameters:

Type	Parameter	Description
AMP_AVENC_ HDLR_s	**hdlr**	Encoder instance. Please see Section 7.9.4.1 for more details.
AMP_VIDEO-ENC_LAYOUT_ CFG_s	**layout**	Current layout..Please refer to Section 7.9.7.1 for more details.

Table 7-140. Parameters for Codec SDK6 API **AmpVideoEnc_GetLayout()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-141. Returns for Codec SDK6 API **AmpVideoEnc_GetLayout()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 21 for more details on error codes.

7.9.7.1 AmpVideoEnc_GetLayout> AMP_VIDEOENC_LAYOUT_CFG_s

Type	Field	Description
UINT16	**Width**	Main Output Width
UINT16	**height**	Main output height
UINT32	**LayerNumber**	Number of layouts
AMP_VIDEO-ENC_LAYER_ DESC_s*	**Layer**	Layout descriptor

*Table 7-142. Definition of **AMP_VIDEOENC_LAYOUT_CFG_s** for Codec SDK6 API **AmpVideoEnc_GetLayout**()*

7.9.7.2 AmpVideoEnc_GetLayout> AMP_VIDEOENC_LAYER_DESC_s

Type	Field	Description
UINT8	**LayerId**	Layer ID
UINT8	**EnableSourceArea**	SourceArea is valid or not. If not, use full source resolution.
UNIT8	**EnableTargetArea**	targetArea is valid or not
AMP_ENC_SOURCE_TYPE_e	**SourceType**	From VIN or memory
Void*	**Source**	From which VIN
UINT32	**SourceLayoutId**	From Nth layout of the source
AMP_AREA_s	**SourceArea**	Area cropped from the source plane
AMP_AREA_s	**TargetArea**	Area within output plane (vinWidth and vinHeight)
AMP_AREA_s	**MultiChanEncSrcArea**	Area within encode input plane
AMP_AREA_s	**MultiChanEncDestArea**	Area within Encode output plane
AMP_2D_BUFFER_s	**MultiChanEncBlendTable**	Blending Table buffer
AMP_ROTATION_e	**SourceRotate**	Rotate after crop
AMBA_DSP_BUF_s	**MemorySource**	Memory starting buffer when VIN comes from memory
UINT16	**MemoryBufferNumber**	Memory buffer total number when VIN comes from memory
UINT8	**MemoryRawBayerPattern**	Raw file bayer pattern when VIN comes from memory
UINT8	**MemoryRawDataBits**	Raw file bits resolution when VIN comes from memory

*Table 7-143. Definition of **AMP_VIDEOENC_LAYER_DESC_s** for Codec SDK6 API **AmpVideoEnc_GetLayout**()*

7.9.8 AmpVideoEnc_SetBitstreamConfig

API Syntax:

AmpVideoEnc_SetBitstreamConfig (AMP_AVENC_HDLR_s * hdlr, AMP_VIDEOENC_BITSTREAM_CFG_s *cfg)

Function Description:

- This function is used to configure bitstream before encoding.

Parameters:

Type	Parameter	Description
AMP_AVENC_HDLR_s	**hdlr**	Encoder instance.. Please see Section 7.9.4.1 for more details.
AMP_VIDEOENC_BIT-STREAM_CFG_s	**cfg**	Configuration. Please see Section 7.9.8.1 for more details.

Table 7-144. Parameters for Codec SDK6 API **AmpVideoEnc_SetBitstreamConfig()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-145. Returns for Codec SDK6 API **AmpVideoEnc_SetBitstreamConfig()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.8.1 AmpVideoEnc_SetBitstreamConfig> AMP_VIDEOENC_BITSTREAM_CFG_s

Type	Field	Description
AMP_VIDEOENC_STREAM_SPEC_e	**StreamSpec**	Stream spec
UINT8	**TimeLapse**	Time lapse encoding, only pri_stream can trigger
UINT8	**VideoThumbnail**	Video thumbnail
AMP_ROTATION_e	**Rotate**	Encode rotation
UNION	**Spec**	Encode specification

*Table 7-146. Definition of **AMP_VIDEOENC_BITSTREAM_CFG_s** for Codec SDK6 API **AmpVideoEnc_SetBitstream-Config**()*

7.9.9 AmpVideoEnc_SetBitstreamBuffer

API Syntax:

AmpVideoEnc_SetBitstreamConfig (AMP_AVENC_HDLR_s * hdlr, AMP_ENC_BITSBUF_CFG_s * cfg)

Function Description:

- This function is used to configure bitstream before encoding.

Parameters:

Type	Parameter	Description
AMP_AVENC_ HDLR_s	**hdlr**	Encoder instance. Please see Section 7.9.4.1 for more details.
AMP_ENC_ BITSBUF_ CFG_s	**cfg**	Buffer configuration. Please see Section 7.9.9.1 for more details.

Table 7-147. Parameters for Codec SDK6 API **AmpVideoEnc_SetBitstreamBuffer()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-148. Returns for Codec SDK6 API **AmpVideoEnc_SetBitstreamBuffer()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.9.1 AmpVideoEnc_SetBitstreamBuffer> AMP_ENC_BITSBUF_CFG_s

Type	Field	Description
UINT8*	**BitsBufAddr**	Bitstream buffer address
UINT32	**BitsBufSize**	Bitstream buffer size. Must be multiple of 16384
UINT8*	**DescBufAddr**	Descriptor buffer address
UINT32	**DescBufSize**	Descriptor buffer size
UINT32	**BitsRunoutThreshold**	Indicates Bitstream's capacity is almost full, Byte Unit

*Table 7-149. Definition of **AMP_ENC_BITSBUF_CFG_s** for Codec SDK6 API **AmpVideoEnc_SetBitstreamBuffer**()*

7.9.10 AmpVideoEnc_GetBitstreamConfig

API Syntax:

AmpVideoEnc_GetBitstreamConfig (AMP_AVENC_HDLR_s * hdlr, AMP_VIDEOENC_BITSTREAM_CFG_s * cfg)

Function Description:

- This function is used to get the current bitstream configuration.

Parameters:

Type	Parameter	Description
AMP_AVENC_HDLR_s	**hdlr**	Encoder instance. Please see Section 7.9.4.1 for more details.
AMP_VIDEOENC_BITSTREAM_CFG_s	**cfg**	Configuration

Table 7-150. Parameters for Codec SDK6 API **AmpVideoEnc_GetBitstreamConfig()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-151. Returns for Codec SDK6 API **AmpVideoEnc_GetBitstreamConfig()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.11 AmpVideoEnc_GetDefaultH264Header

API Syntax:

AmpVideoEnc_GetDefaultCfg (AMP_VIDEOENC_H264_HEADER_INFO_s * info, AMP_VIDEOENC_H264_SPS_s * sps, AMP_VIDEOENC_H264_VUI_s * vui)

Function Description:

- This function is used to get the default H.264 headers based on the current configuration.

Parameters:

Type	Parameter	Description
AMP_VIDEO-ENC_H264_HEADER_INFO_s	**info**	Information for the header generation. Please refer to Section 7.9.11.1 for more details.
AMP_VIDEO-ENC_H264_SPS_s	**sps**	Default SPS header. Please refer to Section 7.9.11.2 for more details.
AMP_VIDEO-ENC_H264_VUI_s	**vui**	Default VUI header

Table 7-152. Parameters for Codec SDK6 API **AmpVideoEnc_GetDefaultH264Header()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-153. Returns for Codec SDK6 API **AmpVideoEnc_GetDefaultH264Header()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.11.1 AmpVideoEnc_GetDefaultH264Header> AMP_VIDEOENC_H264_HEADER_INFO_s

Type	Field	Description
UINT16	**Width**	Clip width
UINT16	**Height**	Clip Height

Type	Field	Description
UINT8	**Interlace**	Interlace
UINT8	**GopM**	Gop M number
AMP_ROTATION_e	**Rotation**	Rotation

*Table 7-154. Definition of **AMP_VIDEOENC_H264_HEADER_INFO_s** for Codec SDK6 API **AmpVideoEnc_GetDefaultH264Header**()*

7.9.11.2 AmpVideoEnc_GetDefaultH264Header> AMP_VIDEOENC_H264_SPS_s

Type	Field	Description
UINT8	**profile_idc**	Please refer to ITU-T H.264
UINT8	**level_idc**	Please refer to ITU-T H.264
UINT8	**frame_cropping_flag**	Please refer to ITU-T H.264
UINT8	**num_ref_frame**	Please refer to ITU-T H.264
UINT16	**frame_crop_left_offset**	Please refer to ITU-T H.264
UINT16	**frame_crop_right_offset**	Please refer to ITU-T H.264
UINT16	**frame_crop_top_offset**	Please refer to ITU-T H.264
UINT16	**frame_crop_bottom_offset**	Please refer to ITU-T H.264
UINT8	**log2_max_frame_num_minus4**	Please refer to ITU-T H.264
UINT8	**log2_max_pic_order_cnt_lsb_minus4**	Please refer to ITU-T H.264

*Table 7-155. Definition of **AMP_VIDEOENC_H264_SPS_s** for Codec SDK6 API **AmpVideoEnc_GetDefaultH264Header**()*

7.9.12 AmpVideoEnc_SetRuntimeBitrate

API Syntax:

AmpVideoEnc_SetRuntimeBitrate (AMP_AVENC_HDLR_s * hdlr, AMP_VIDEOENC_RUNTIME_BITRATE_CFG_s * cfg)

Function Description:

- This function is used in runtime to control bitrate during the video recording state.

Parameters:

Type	Parameter	Description
AMP_AVENC_HDLR_s	**hdlr**	Encoder instance. Please see Section 7.9.4.1 for more details.
AMP_VIDEOENC_RUNTIME_BITRATE_CFG_s	**cfg**	Runtime control configuration. Please see Section 7.9.12.1 for more details.

Table 7-156. Parameters for Codec SDK6 API **AmpVideoEnc_SetRuntimeBitrate()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-157. Returns for Codec SDK6 API **AmpVideoEnc_SetRuntimeBitrate()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.12.1 AmpVideoEnc_SetRuntimeBitrate> AMP_VIDEOENC_RUNTIME_BITRATE_CFG_s

Type	Field	Description
UINT32	**AverageBitrate**	Average bitrate in Byte unit

*Table 7-158. Definition of **AMP_VIDEOENC_RUNTIME_BITRATE_CFG_s** for Codec SDK6 API **AmpVideoEnc_SetRuntimeBitrate**()*

7.9.13 AmpVideoEnc_SetRuntimeQuality

API Syntax:

AmpVideoEnc_SetRuntimeQuality (AMP_AVENC_HDLR_s * hdlr, AMP_VIDEOENC_RUNTIME_QUALITY_CFG_s * cfg)

Function Description:

- This function is used in runtime to control quality during the video recording state.

Parameters:

Type	Parameter	Description
AMP_AVENC_HDLR_s	**hdlr**	Encoder instance. Please see Section 7.9.4.1 for more details.
AMP_VIDEO-ENC_RUN-TIME_QUAL-ITY_CFG_s	**cfg**	Runtime control configuration. Please refer to Section 7.9.13.1 for more details.

Table 7-159. Parameters for Codec SDK6 API **AmpVideoEnc_SetRuntimeQuality()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-160. Returns for Codec SDK6 API **AmpVideoEnc_SetRuntimeQuality()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.13.1 AmpVideoEnc_SetRuntimeQuality> AMP_VIDEOENC_RUNTIME_QUALITY_CFG_s

Type	Field	Description
UINT32	**Cmd**	Quality control command #define RC_BITRATE (1<<0) #define RC_GOP (1<<1) #define RC_QP (1<<2) #define RC_QMODEL (1<<3) #define RC_ROI (1<<4) #define RC_HQP (1<<5) #define RC_ZMV (1<<6) #define RC_FORCE_IDR (1<<7)
UINT32	**BitRate**	Desired bit rate
UINT8	**Enable**	Enable or not
UINT8	**M**	Gop structure M
UINT8	**N**	Gop structure N
UINT8	**IDR**	Gop structure IDR
UINT8	**QpMinI**	QP min value on I frame
UINT8	**QpMaxI**	QP max value on I frame
UINT8	**QpMinP**	QP min value on P frame
UINT8	**QpMaxP**	QP max value on P frame
UINT8	**QpMinB**	QP min value on B frame
UINT8	**QpMaxB**	QP max value on B frame
UINT8	**HQpMax**	High quality P frame max QP value
UINT8	**HQpMin**	High quality P frame min QP value
UINT8	**HQpReduce**	High quality P frame reduce number
UINT8	**HPNumber**	High quality P frame number
UINT8	**IorIDRFrameNeedsRateControlMask**	Please refer to AQP control
UINT8	**QPReduceNearIDRFrame**	Please refer to AQP control
UINT8	**ClassNumberLimitOfQPReduceNearIDRFrame**	Please refer to AQP control
UINT8	**QPReduceNearIFrame**	Please refer to AQP control
UINT8	**ClassNumberLimitOfQPReduceNearIFrame**	Please refer to AQP control
INT8	**Intra16x16Bias**	-64 ~ 64
INT8	**Intra4x4Bias**	-64 ~ 64
INT8	**Inter16x16Bias**	-64 ~ 64
INT8	**Inter8x8Bias**	-64 ~ 64
INT8	**Direct16x16Bias**	-64 ~ 64
INT8	**Direct8x8Bias**	-64 ~ 64
INT8	**MELambdaQpOffset**	-64 ~ 64
INT8	**AQPStrength**	0 - auto, 1-81: fixed strength, 1 for no AQP, -1 for inverse AQP
INT8	**LoopFilterAlpha**	-6 ~ 6
INT8	**LoopFilterBeta**	-6 ~ 6
INT8	**Alpha**	Please refer to AQP control
INT8	**Beta**	Please refer to AQP control

Type	Field	Description
UINT32*	**RoiBufferAddr**	Region of interset buffer address
INT8	**RoiDelta[3][4]**	Region of interset definition array
UINT32	**ZmvThres**	Zero motion vector threshold

*Table 7-161. Definition of **AMP_VIDEOENC_RUNTIME_QUALITY_CFG_s** for Codec SDK6 API **AmpVideoEnc_SetRuntimeQuality**()*

7.9.14 AmpVideoEnc_CaptureTimeLapsedFrame

API Syntax:

AmpVideoEnc_CaptureTimeLapsedFrame (AMP_VIN_HDLR_s * hdlr)

Function Description:

- This function is used to capture and encode one frame when there is time-lapsed recoding.

Parameters:

Type	Parameter	Description
AMP_VIN_ HDLR_s	**hdlr**	Vin instance. Please see Section 8.2.4.1 for more details.

Table 7-162. Parameters for Codec SDK6 API **AmpVideoEnc_CaptureTimeLapsedFrame()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-163. Returns for Codec SDK6 API **AmpVideoEnc_CaptureTimeLapsedFrame()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.15 AmpVideoEnc_SetRuntimeFrameRate

API Syntax:

AmpVideoEnc_SetRuntimeFrameRate (UINT32 numHdlr, AMP_VIDEOENC_RUNTIME_FRAMERATE_CFG_s * cfg)

Function Description:

- This function is used to change framerate in runtime.

Parameters:

Type	Parameter	Description
UINT32	**numHdlr**	Number of handlers to change
AMP_VIDEOENC_RUNTIME_FRAMERATE_CFG_s	**cfg**	Handler and configuration array. Please refer to Section 7.9.15.1 for more details.

Table 7-164. Parameters for Codec SDK6 API **AmpVideoEnc_SetRuntimeFrameRate()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-165. Returns for Codec SDK6 API **AmpVideoEnc_SetRuntimeFrameRate()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.15.1 AmpVideoEnc_SetRuntimeFrameRate> AMP_VIDEOENC_RUNTIME_FRAMERATE_CFG_s

Type	Field	Description
AMP_AVENC_HDLR_s*	**Hdlr**	Point to which video encode handler
UINT32	**Divisor**	Frame rate divisor

Table 7-166. Definition of ***AMP_VIDEOENC_RUNTIME_FRAMERATE_CFG_s*** *for Codec SDK6 API* ***AmpVideoEnc_SetRuntimeFrameRate****()*

7.9.16 AmpVideoEnc_SetEncodeBlend

API Syntax:

AmpVideoEnc_SetEncodeBlend (AMP_AVENC_HDLR_s * hdlr, AMP_VIDEOENC_BLEND_INFO_s * info)

Function Description:

- This function is used to set the encode blend information.

Parameters:

Type	Parameter	Description
AMP_AVENC_HDLR_s	**hdlr**	Encoder instance. Please see Section 7.9.4.1 for more details.
AMP_VIDEO-ENC_BLEND_INFO_s	**info**	Blend information. Please refer to Section 7.9.16.1 for more details.

Table 7-167. Parameters for Codec SDK6 API **AmpVideoEnc_SetEncodeBlend()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-168. Returns for Codec SDK6 API **AmpVideoEnc_SetEncodeBlend()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.16.1 AmpVideoEnc_SetEncodeBlend> AMP_VIDEOENC_BLEND_INFO_s

Type	Field	Description
UINT8	**BufferID**	Blend buffer ID, 0~ 31 (at most 32 blending areas)
UINT8	**Enable**	0: Disable, 1: Enable
UINT16	**OffsetX**	Horizontal offset with respect to main picture
UINT16	**OffsetY**	Vertical offset with respect to main picture
UINT16	**Pitch**	Blend buffer pitch
UINT16	**Width**	Blend buffer width
UINT16	**Height**	Blend buffer height
UINT8*	**YAddr**	Blend buffer luma address

Type	Field	Description
UINT8*	**UVAddr**	Blend buffer chroma address
UINT8*	**AlphaYAddr**	Alpha plane luma address
UINT8*	**AlphaUVAddr**	Alpha plane chroma address

*Table 7-169. Definition of **AMP_VIDEOENC_BLEND_INFO_s** for Codec SDK6 API **AmpVideoEnc_SetEncodeBlend**()*

7.9.17 AmpVideoEnc_FeedEncodeRaw

API Syntax:

AmpVideoEnc_FeedEncodeRaw (AMP_ENC_RAW_INFO_s * raw, UINT8 lastFrame)

Function Description:

- This function is used to feed RAW/YUV for video RawEncode.

Parameters:

Type	Parameter	Description
AMP_ENC_ RAW_INFO_s	**raw**	Raw buffer information. Please refer to Section 7.9.17.1 for more details.
UINT8	**lastFrame**	Indicates the last encode frame

Table 7-170. Parameters for Codec SDK6 API **AmpVideoEnc_FeedEncodeRaw().**

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-171. Returns for Codec SDK6 API **AmpVideoEnc_FeedEncodeRaw().**

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.17.1 AmpVideoEnc_FeedEncodeRaw> AMP_ENC_RAW_INFO_s

Type	Field	Description
UINT8	**compressed**	CFA Raw buffer format
UINT8*	**RawAddr**	CFA Raw buffer address
UINT16	**RawPitch**	CFA Raw buffer pitch
UINT16	**RawWidth**	CFA Raw buffer width
UINT16	**RawHeight**	CFA Raw buffer height
UINT16	**OBWidth**	CFA Raw buffer OB region width
UINT16	**OBHeight**	CFA Raw buffer OB region height
UINT16	**OBOffsetX**	CFA Raw buffer OB region offset in x-axis
UINT16	**OBOffsetY**	CFA Raw buffer OB region offset in y-axis

*Table 7-172. Definition of **AMP_ENC_RAW_INFO_s** for Codec SDK6 API **AmpVideoEnc_FeedEncodeRaw**()*

7.9.18 AmpVideoEnc_GetEncodingInfo

API Syntax:

AmpVideoEnc_GetEncodingInfo (AMP_AVENC_HDLR_s * hdlr, AMP_VIDEOENC_ENCODING_INFO_s * info)

Function Description:

- This function is used to get the encoding session information.

Parameters:

Type	Parameter	Description
AMP_AVENC_HDLR_s	**hdlr**	Encoder instance. Please see Section 7.9.4.1 for more details.
AMP_VIDEOENC_ENCODING_INFO_s	**info**	Encoding information. Please see Section 7.9.18.1 for more details.

Table 7-173. Parameters for Codec SDK6 API **AmpVideoEnc_GetEncodingInfo()**.

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 7-174. Returns for Codec SDK6 API **AmpVideoEnc_GetEncodingInfo()**.

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

7.9.18.1 AmpVideoEnc_GetEncodingInfo> AMP_VIDEOENC_ENCODING_INFO_s

Type	Field	Description
UINT32	**TotalFrames**	Total encoded frames until now
UINT32	**AverageBitrate**	Average bitrate until now, unit: kbps
UINT64	**TotalBytes**	Total encoded bytes until now, unit: Bytes

*Table 7-175. Definition of **AMP_VIDEOENC_ENCODING_INFO_s** for Codec SDK6 API **AmpVideoEnc_GetEncodingInfo**()*

8 VIN

8.1 VIN: Overview

This chapter provides information on the VIN manager implementation. When the user implements the VIN manager/handler, the user is able to initialize the VIN manager function, create the VIN manager function, delete the VIN manager function, and configure the VIN manager after the VIN manager has been created. For details on these functions, please refer to the sections below.

8.2 VIN: List of APIs

This section lists the VIN API functions.

- AmpVin_GetInitDefaultCfg
- AmpVin_Init
- AmpVin_GetDefaultCfg
- AmpVin_Create
- AmpVin_Delete
- AmpVin_ConfigHandler

8.2.1 AmpVin_GetInitDefaultCfg

API Syntax:

AmpVin_GetInitDefaultCfg (AMP_VIN_INIT_CFG_s * defaultCfg)

Function Description:

- This function gets the default VIN configuration values to start initialization.

Parameters:

Type	Parameter	Description
AMP_VIN_INIT_CFG_s	**defaultCfg**	Default values for VIN initialization. (**AMP_VIN_INIT_CFG_s** is defined in `vin.h`) Please refer to Section 8.2.1.1 below for more details.

*Table 8-1. Parameters for VIN SDK6 API **AmpVin_GetInitDefaultCfg()**.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 8-2. Returns for VIN SDK6 API **AmpVin_GetInitDefaultCfg()**.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

8.2.1.1 AmpVin_GetInitDefaultCfg > AMP_VIN_INIT_CFG_s

Type	Field	Description
UINT32	**NumMaxVin**	Maximum VIN number
UINT8 *	**MemoryPoolAddr**	Start Address of the buffer for the VIN module
UINT32	**MemoryPoolSize**	Size of the buffer

*Table 8-3. Definition of **AMP_VIN_INIT_CFG_s** for VIN SDK6 API **AmpVin_GetInitDefaultCfg**().*

8.2.2 AmpVin_Init

API Syntax:

AmpVin_Init (AMP_VIN_INIT_CFG_s * cfg)

Function Description:

- This function initializes the VIN module. The function should only be invoked once. The user MUST invoke this function before using the VIN module. The memory pool of the module will be provided by the user.

Parameters:

Type	Parameter	Description
AMP_VIN_INIT_CFG_s	**Cfg**	Default values for VIN instance configuration (**AMP_VIN_INIT_CFG_s** is defined in `vin.h`). Please refer to Section 8.2.1.1 for more details.

*Table 8-4. Parameters for VIN SDK6 API **AmpVin_Init()**.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 8-5. Returns for VIN SDK6 API **AmpVin_Init()**.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

8.2.3 AmpVin_GetDefaultCfg

API Syntax:

AmpVin_GetDefaultCfg (AMP_VIN_HDLR_CFG_s * cfg)

Function Description:

- This function gets the default VIN instance configuration.

Parameters:

Type	Parameter	Description
AMP_VIN_ HDLR_CFG_s	**Cfg**	Default Values for individual VIN instance configuration. (**AMP_VIN_HDLR_CFG_s** is defined in `vin.h`) Please refer to Section 8.2.3.1 below for more details.

*Table 8-6. Parameters for VIN SDK6 API **AmpVin_GetDefaultCfg()**.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 8-7. Returns for VIN SDK6 API **AmpVin_GetDefaultCfg()**.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

8.2.3.1 AmpVin_GetDefaultCfg > AMP_VIN_HDLR_CFG_s

Type	Field	Description
AMBA_DSP_CHANNEL_ID_u	**Channel**	Hardware device description
AMBA_SENSOR_MODE_ID_u	**Mode**	Mode ID
AMBA_YUV_MODE_ID_u	**YuvMode**	YUV mode ID
UINT16	**LayoutNumber**	Number of main windows that are produced from this VIN
AMP_VIN_LAYOUT_CFG_s *	**Layout**	Main window
AMP_AREA_s	**HwCaptureWindow**	Hardware capture window
AMP_CALLBACK_f	**cbSwitch**	Mode switch callback
AMP_CALLBACK_f	**cbEvent**	General event callback
AMP_CALLBACK_f	**cbCfgUpdated**	TBD
AMP_TASK_INFO_s	**TaskInfo**	Task information
AMP_VIN_SOURCE_e	**VinSource**	VIN Source

*Table 8-8. Definition of **AMP_VIN_HDLR_CFG_s** for VIN SDK6 API **AmpVin_GetDefaultCfg**().*

8.2.4 AmpVin_Create

API Syntax:

AmpVin_Create (AMP_VIN_HDLR_CFG_s * cfg, AMP_VIN_HDLR_s ** vinHdlr **)**

Function Description:

- This function creates and sets the initial configuration of a VIN handler.

Parameters:

Type	Parameter	Description
AMP_VIN_ HDLR_CFG_s	**Cfg**	Default values for individual VIN instance configuration (**AMP_VIN_HDLR_CFG_s** is defined in `vin.h`). Please refer to Section 8.2.3.1 for more details.
AMP_VIN_ HDLR_s	**vinHdlr**	Intializes the VIN Instance (**AMP_VIN_HDLR_s** is defined in `vin.h`). Please refer to Section 8.2.4.1 below for more details.

*Table 8-9. Parameters for VIN SDK6 API **AmpVin_Create()**.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 8-10. Returns for VIN SDK6 API **AmpVin_Create()**.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

8.2.4.1 AmpVin_Create > AMP_VIN_HDLR_s

Type	Field	Description
void*	**Ctx**	Private context of the VIN

*Table 8-11. Definition of **AMP_VIN_HDLR_s** for VIN SDK6 API **AmpVin_Create**().*

8.2.5 AmpVin_Delete

API Syntax:

AmpVin_Delete (AMP_VIN_HDLR_s * vinHdlr)

Function Description:

- This function deletes the VIN handler.

Parameters:

Type	Parameter	Description
AMP_VIN_ HDLR_s	**vinHdlr**	Deletes the VIN handler instance (**AMP_VIN_HDLR_s** is defined in `vin.h`). Please refer to Section 8.2.4.1 for more details.

*Table 8-12. Parameters for VIN SDK6 API **AmpVin_Delete()**.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

*Table 8-13. Returns for VIN SDK6 API **AmpVin_Delete()**.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

8.2.6 AmpVin_ConfigHandler

API Syntax:

AmpVin_ConfigHandler (AMP_VIN_HDLR_s * vinHdlr, AMP_VIN_CFG_s * vinCfg)

Function Description:

- This function changes the VIN settings of a VIN handler when liveview is not running.

Parameters:

Type	Parameter	Description
AMP_VIN_ HDLR_s	**vinHdlr**	Default values for VIN handler instance (**AMP_VIN_HDLR_s** is defined in `vin.h`) Please refer to Section 8.2.4.1 for more details.
AMP_VIN_ CFG_s	**vinCfg**	Offline VIN configurations for a given handler (when liveview is not running) (**AMP_VIN_CFG_s** is defined in `vin.h`) Please refer to Section 8.2.6.1 below for more details.

Table 8-14. Parameters for VIN SDK6 API ***AmpVin_ConfigHandler()****.*

Returns:

Return	Description
AMP_OK	Execution Successful
All Others	AMP_ER_CODE_e. Please refer to Chapter 10 for more details on error codes.

Table 8-15. Returns for VIN SDK6 API ***AmpVin_ConfigHandler()****.*

Example:

Please refer to the Unit Test document.

See Also:

Please refer to Chapter 10 for more details on error codes.

8.2.6.1 AmpVin_ConfigHandler > AMP_VIN_CFG_s

Type	Field	Description
AMBA_DSP_CHANNEL_ID_u	**Channel**	Hardware device description
AMBA_SENSOR_MODE_ID_u	**Mode**	Mode ID
AMBA_YUV_MODE_ID_u	**YuvMode**	YUV mode ID
UINT16	**LayoutNumber**	The number of main windows produced from this VIN.
AMP_VIN_LAYOUT_CFG_s *	**Layout**	Main window
AMP_AREA_s	**HwCaptureWindow**	Hardware capture window
AMP_CALLBACK_f	**cbSwitch**	Mode Switch callback
AMP_CALLBACK_f	**cbEvent**	General Event Callback
AMP_CALLBACK_f	**cbCfgUpdated**	TBD
AMP_VIN_SOURCE_e	**VinSource**	Vin Source

*Table 8-16. Definition of **AMP_VIN_CFG_s** for VIN SDK6 API **AmpVin_ConfigHandler**().*

8.2.6.1.1 AmpVin_ConfigHandler > AMP_VIN_CFG_s > AMP_VIN_LAYOUT_CFG_s

Type	Field	Description
UINT8:1	**EnableSourceArea**	SourceArea is valid or not
UINT8:1	**EnableOBArea**	OBArea is valid or not
UINT8:6	**AreaEnReserved**	
AMP_ROTATION_e	**Rotation**	Rotate after crop
AMP_AREA_s	**SourceArea**	Area cropped from the source plane
AMP_AREA_s	**ActiveArea**	Area of Active cropeed from the Vcap plane
AMP_AREA_s	**OBArea**	Area of OB cropped from the Vcap plane
UINT16	**Width**	IDSP output width (main window) equal to codec's source plane
UINT16	**Height**	IDSP output height (main window) equal to codec's source plane
int	**(*cfaOutputCallback)(AMBA_DSP_IMG_WARP_CALC_CFA_INFO_s*pCfaInfo)**	Customer's CFA output calculation callback
UINT32	**DzoomFactorX**	Initial horizontal digital zoom factor, 16.16 format
UINT32	**DzoomFactorY**	Initial vertical digital zoom factor, 16.16 format
INT32	**DzoomOffsetX**	Initial horizontal digital zoom offset, 16.16 format
INT32	**DzoonOffsetY**	Initial vertical digital zoom offset, 16.16
UINT8	**MainviewReportRate**	Mainview status report rate

*Table 8-17. Definition of **AMP_VIN_LAYOUT_CFG_s** for VIN SDK6 API **AmpVin_ConfigHandler**().*

8.2.6.1.2 AmpVin_ConfigHandler > AMP_VIN_CFG_s > AMP_VIN_SOURCE_e

Type	Field	Description
Enum	**AMP_VIN_EXTERNAL_CFA**	VIN signal comes from an external CFA output, i.e. sensor
	AMP_VIN_EXTERNAL_YUV	VIN signal comes from external YUV output, i.e. DVR
	AMP_VIN_MEMORY_CFA	VIN signal comes from memory CFA
	AMP_VIN_MEMORY_YUV	VIN signal comes from memory YUV

*Table 8-18. Definition of **AMP_VIN_SOURCE_e** for VIN SDK6 API **AmpVin_ConfigHandler**().*

9 System Errors

9.1 System Errors: Overview

This chapter lists the possible return values (**AMP**) when errors are encountered.

9.2 System Errors: Error Code List

- AMP_ERROR_GENERAL_ERROR
- AMP_ERROR_INCORRECT_PARAM_STRUCTURE
- AMP_ERROR_INCORRECT_PARAM_VALUE_RANGE
- AMP_ERROR_OUT_OF_MEMORY
- AMP_ERROR_RESOURCE_INVALID
- AMP_ERROR_FIFO_TYPE_MISMATCH
- AMP_ERROR_FIFO_LOCKED
- AMP_ERROR_FIFO_EMPTY
- AMP_ERROR_FIFO_FULL
- AMP_ERROR_ILLEGAL_OPERATION
- AMP_ERROR_ILLEGAL_CONTAIN_SOURCE
- AMP_ERROR_IO_ERROR
- AMP_ERROR_OUT_OF_STORAGE
- AMP_ERROR_OPERATION_ABORTED
- AMP_ERROR_OBJ_ALREADY_EXISTS
- AMP_ERROR_OBJ_UNAVAILABLE
- AMP_ERROR_OBJ_CREATION_FAILED

9.2.1 AMP_ERROR_GENERAL_ERROR

Error Value:

AMP_ERROE_GENERAL_ERROR

Error Description:

- General error.

9.2.2 AMP_ERROR_INCORRECT_PARAM_STRUCTURE

Error Value:

AMP_ERROR_INCORRECT_PARAM_STRUCTURE

Error Description:

- Incorrect structure used.

9.2.3 AMP_ERROR_INCORRECT_PARAM_VALUE_RANGE

Error Value:

AMP_ERROR_INCORRECT_PARAM_VALUE_RANGE

Error Description:

- Incorrect value range.

9.2.4 AMP_ERROR_OUT_OF_MEMORY

Error Value:

AMP_ERROR_OUT_OF_MEMORY

Error Description:

- Out of memory.

9.2.5 AMP_ERROR_RESOURCE_INVALID

Error Value:

AMP_ERROR_RESOURCE_INVALID

Error Description:

- Resource for the operation.

9.2.6 AMP_ERROR_FIFO_TYPE_MISMATCH

Error Value:

AMP_ERROR_FIFO_TYPE_MISMATCH

Error Description:

- Incorrect FIFO type.

9.2.7 AMP_ERROR_FIFO_LOCKED

Error Value:

AMP_ERROR_FIFO_LOCKED

Error Description:

- Try to read/write a locked FIFO.

9.2.8 AMP_ERROR_FIFO_EMPTY

Error Value:

AMP_ERROR_FIFO_EMPTY

Error Description:

- No entry in the FIFO.

9.2.9 AMP_ERROR_FIFO_FULL

Error Value:

AMP_ERROR_FULL

Error Description:

- FIFO full.

9.2.10 AMP_ERROR_ILLEGAL_OPERATION

Error Value:

AMP_ERROR_ILLEGAL_OPERATIOM

Error Description:

- Illegal operation.

9.2.11 AMP_ERROR_ILLEGAL_CONTAIN_SOURCE

Error Value:

AMP_ERROR_ILLEGAL_CONTAIN_SOURCE

Error Description:

- Illegal container source.

9.2.12 AMP_ERROR_IO_ERROR

Error Value:

AMP_ERROR_IO_ERROR

Error Description:

- Stream IO error.

9.2.13 AMP_ERROR_OUT_OF_STORAGE

Error Value:

AMP_ERROR_OUT_OF_STORAGE

Error Description:

- Storage media full.

9.2.14 AMP_ERROR_OPERATION_ABORTED

Error Value:

AMP_ERROR_IO_OPERATION_ABORTED

Error Description:

- Operation aborted after a user aborts it.

9.2.15 AMP_ERROR_OBJ_ALREADY_EXISTS

Error Value:

AMP_ERROR_OBJ_ALREADY_EXISTS

Error Description:

- An object has already existed.

9.2.16 AMP_ERROR_OBJ_UNAVAILABLE

Error Value:

AMP_ERROR_OBJ_UNAVAILABLE

Error Description:

- A request object is unavailable.

9.2.17 AMP_ERROR_OBJ_CREATION_FAILED

Error Value:

AMP_ERROR_OBJ_CREATION_FAILED

Error Description:

- Failed to create an object.

Appendix 2 Important Notice

All Ambarella design specifications, datasheets, drawings, files, and other documents (together and separately, “materials”) are provided on an “*as is*” basis, and Ambarella makes no warranties, expressed, implied, statutory, or otherwise with respect to the materials, and expressly disclaims all implied warranties of noninfringement, merchantability, and fitness for a particular purpose. The information contained herein is believed to be accurate and reliable. However, Ambarella assumes no responsibility for the consequences of use of such information.

Ambarella Incorporated reserves the right to correct, modify, enhance, improve, and otherwise change 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 Ambarella’s terms and conditions of sale supplied at the time of order acknowledgment. Ambarella warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with its standard warranty. Testing and other quality control techniques are used to the extent Ambarella deems necessary to support this warranty.

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

Ambarella does not warrant or represent that any license, either expressed or implied, is granted under any Ambarella patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which Ambarella products or services are used. Information published by Ambarella regarding third-party products or services does not constitute a license from Ambarella 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 Ambarella under the patents or other intellectual property of Ambarella.

Reproduction of information from Ambarella documents is not permissible without prior approval from Ambarella.

Ambarella products are not authorized for use in safety-critical applications (such as life support) where a failure of the product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Customers acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of Ambarella products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by Ambarella. Further, Customers must fully indemnify Ambarella and its representatives against any damages arising out of the use of Ambarella products in such safety-critical applications.

Ambarella products are neither designed nor intended for use in automotive and military/aerospace applications or environments. Customers acknowledge and agree that any such use of Ambarella products is solely at the Customer’s risk, and they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

Appendix 11 Revision History

NOTE: Page numbers for previous drafts may differ from page numbers in the current version.

Version	Date	Comments
1.0	4 November 2014	Preliminary Release
1.1	17 February 2015	Updated Chapters 5, 6, 7, 8, 9 API syntax, parameters and/or register information; Added Section 1.2.18
1.2	24 June 2015	Updated Sections 1.2.1.2, 5.2.5.1, 7.9.14, 8.8.3.1, and 8.9.14 Added Sections 1.2.1.3 and 3.2.12 Deleted Chapter 5

Table A11-1. Revision History.