

# SDK6 API Middleware (Service)

Version 1.3

June 23, 2015



Confidentiality Notice:

Copyright © 2015 Ambarella Inc.

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

<b>1</b>	<b>Cached File System (CFS)</b>	<b>1</b>
1.1	CFS: Overview	1
1.2	CFS: List of Functions	1
<b>2</b>	<b>Design Rule for Camera File System</b>	<b>39</b>
2.1	DCF: Overview	39
2.2	DCF: List of Functions	39
<b>3</b>	<b>Demuxer</b>	<b>72</b>
3.1	Demuxer: Overview	72
3.2	Demuxer: List of Functions	72
<b>4</b>	<b>Editor</b>	<b>99</b>
4.1	Editor: Overview	99
4.2	Editor: List of APIs	99
<b>5</b>	<b>Format</b>	<b>110</b>
5.1	Format: Overview	110
5.2	Format: List of APIs	110
5.3	Dummy	143
5.4	EXIF	152
5.5	EXT	169
5.6	Matroska	184
5.7	MOV	204
5.8	MP4	226
<b>6</b>	<b>Muxer</b>	<b>249</b>
6.1	Muxer: Overview	249
6.2	Muxer: List of APIs	249
<b>7</b>	<b>Index</b>	<b>279</b>
7.1	Index: Overview	279
7.2	Index: List of APIs	279
<b>8</b>	<b>Stream</b>	<b>305</b>

8.1	Stream: Overview .....	305
8.2	Stream: List of APIs .....	305
9	Utility .....	314
9.1	Utility: Overview .....	314
9.2	Utility: List of APIs .....	314
10	System Errors .....	331
10.1	System Errors: Overview .....	331
10.2	System Errors: Error Code List .....	331
Appendix 11	Important Notice .....	A336
Appendix 12	Revision History .....	A337

Confidential  
For HAOTEK Only

# 1 Cached File System (CFS)

## 1.1 CFS: Overview

This chapter provides information regarding the implementation of the CFS module, including the following functionality:

1. CFS read/write
2. CFS seek
3. CFS sync
4. Other file system-related functions

## 1.2 CFS: List of Functions

- (Section 1.2.1) [AmpCFS\\_Cdelete](#)
- (Section 1.2.2) [AmpCFS\\_Chmod](#)
- (Section 1.2.3) [AmpCFS\\_Cinsert](#)
- (Section 1.2.4) [AmpCFS\\_ClearCache](#)
- (Section 1.2.5) [AmpCFS\\_Combine](#)
- (Section 1.2.6) [AmpCFS\\_Divide](#)
- (Section 1.2.7) [AmpCFS\\_fappend](#)
- (Section 1.2.8) [AmpCFS\\_fclose](#)
- (Section 1.2.9) [AmpCFS\\_feof](#)
- (Section 1.2.10) [AmpCFS\\_FGetLen](#)
- (Section 1.2.11) [AmpCFS\\_FirstDirEnt](#)
- (Section 1.2.12) [AmpCFS\\_fopen](#)
- (Section 1.2.13) [AmpCFS\\_Format](#)
- (Section 1.2.14) [AmpCFS\\_fread](#)
- (Section 1.2.15) [AmpCFS\\_fseek](#)
- (Section 1.2.16) [AmpCFS\\_FStatus](#)
- (Section 1.2.17) [AmpCFS\\_FSync](#)
- (Section 1.2.18) [AmpCFS\\_ftell](#)
- (Section 1.2.19) [AmpCFS\\_fwrite](#)
- (Section 1.2.20) [AmpCFS\\_GetCachedDataSize](#)
- (Section 1.2.21) [AmpCFS\\_GetDefaultCfg](#)
- (Section 1.2.22) [AmpCFS\\_GetDev](#)

- (Section 1.2.23) AmpCFS\_GetError
- (Section 1.2.24) AmpCFS\_GetFileMaxCachedDataSize
- (Section 1.2.25) AmpCFS\_GetFileParam
- (Section 1.2.26) AmpCFS\_GetRequiredBufferSize
- (Section 1.2.27) AmpCFS\_Init
- (Section 1.2.28) AmpCFS\_Mkdir
- (Section 1.2.29) AmpCFS\_Mount
- (Section 1.2.30) AmpCFS\_Move
- (Section 1.2.31) AmpCFS\_NextDirEnt
- (Section 1.2.32) AmpCFS\_remove
- (Section 1.2.33) AmpCFS\_Rmdir
- (Section 1.2.34) AmpCFS\_Stat
- (Section 1.2.35) AmpCFS\_Sync
- (Section 1.2.36) AmpCFS\_Unmount

Confidential  
For HAOTEK Only

## 1.2.1 AmpCFS\_Cdelete

### API Syntax:

**AmpCFS\_Cdelete** ( const char \* fileName, UINT32 offset, UINT32 number )

### Function Description:

- This function is used to delete clusters from a file.

### Parameters:

Type	Parameter	Description
const char *	<b>fileName</b>	File path
UINT32	<b>offset</b>	The offset from the beginning of the file to a specified location to delete clusters.
UINT32	<b>number</b>	Number of clusters to be deleted

Table 1-1. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Cdelete()**.

### Returns:

Return	Description
Number of clusters	The number of clusters that are actually deleted

Table 1-2. Returns for SDK6 Middleware CFS API **AmpCFS\_Cdelete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.2 AmpCFS\_Chmod

### API Syntax:

**AmpCFS\_Chmod** (const char \* fileName, int attribute )

### Function Description:

- This function is used to change the attributes of a file.

### Parameters:

Type	Parameter	Description
const char *	<b>fileName</b>	File path
int	<b>attribute</b>	File attributes

Table 1-3. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Chmod()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-4. Returns for SDK6 Middleware CFS API **AmpCFS\_Chmod()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



### 1.2.3 AmpCFS\_Cinsert

#### API Syntax:

**AmpCFS\_Cinsert** ( const char \* fileName, UINT32 offset, UINT32 number )

#### Function Description:

- This function is used to allocate and insert clusters to a file

#### Parameters:

Type	Parameter	Description
const char *	<b>fileName</b>	File Path
UINT32	<b>offset</b>	The offset from the beginning of the file to a specified location to insert clusters.
UINT32	<b>number</b>	Number of clusters being inserted

Table 1-5. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Cinsert()**.

#### Returns:

Return	Description
Number of clusters	Number of clusters that are actually inserted

Table 1-6. Returns for SDK6 Middleware CFS API **AmpCFS\_Cinsert()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.4 AmpCFS\_ClearCache

### API Syntax:

**AmpCFS\_ClearCache** (char driveName)

### Function Description:

- This function is used to clear drive caches which include file states, directory entries and drive information.

### Parameters:

Type	Parameter	Description
char	<b>driveName</b>	Drive name (from A to Z)

Table 1-7. Parameters for SDK6 API Middleware CFS API **AmpCFS\_ClearCache()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-8. Returns for SDK6 Middleware CFS API **AmpCFS\_ClearCache()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.5 AmpCFS\_Combine

### API Syntax:

**AmpCFS\_Combine** ( const char \* baseFilename, const char \* addFilename )

### Function Description:

- This function is used to combine two files into one file.

### Parameters:

Type	Parameter	Description
const char *	<b>baseFilename</b>	Path of the base file
const char *	<b>addFilename</b>	Path of the add file

Table 1-9. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Combine()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-10. Returns for SDK6 Middleware CFS API **AmpCFS\_Combine()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.6 AmpCFS\_Divide

### API Syntax:

**AmpCFS\_Divide** ( const char \* orgFilename, const char \* newFilename, UINT32 offset )

### Function Description:

- This function is used to divide a file into two files.

### Parameters:

Type	Parameter	Description
const char *	<b>orgFilename</b>	Path of a file being divided
const char *	<b>newFilename</b>	Path of a file being created after division is completed
UINT32	<b>offset</b>	Byte offset from the beginning of the original file to a division location.

Table 1-11. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Divide()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-12. Returns for SDK6 Middleware CFS API **AmpCFS\_Divide()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.7 AmpCFS\_fappend

### API Syntax:

**AmpCFS\_fappend** ( AMP\_CFS\_FILE\_s \* file, UINT64 size )

### Function Description:

- This function is used to add consecutive clusters of a specified size to the end of a file.

### Parameters:

Type	Parameter	Description
AMP_CFS_FILE_s *	<b>file</b>	The file descriptor ( <a href="#">Section 1.2.7.1</a> )
UINT64	<b>size</b>	Size of the area to be added (bytes)

Table 1-13. Parameters for SDK6 API Middleware CFS API **AmpCFS\_fappend()**.

### Returns:

Return	Description
Size	The size (bytes) of the added area

Table 1-14. Returns for SDK6 Middleware CFS API **AmpCFS\_fappend()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 1.2.7.1 AmpCFS\_fappend > AMP\_CFS\_FILE\_s

Type	Field	Description
char [MAX_FILE-NAME_LENGTH]	<b>Filename</b>	Full path of a file

Table 1-15. Definition of **AMP\_CFS\_FILE\_s** for CFS API **AmpCFS\_fappend()**.

Confidential  
For HAOTEK Only

## 1.2.8 AmpCFS\_fclose

### API Syntax:

**AmpCFS\_fclose** ( AMP\_CFS\_FILE\_s \* file )

### Function Description:

- This function is used to close a CFS file.

### Parameters:

Type	Parameter	Description
AMP_CFS_FILE_s *	<b>file</b>	CFS file descriptor ( <a href="#">Section 1.2.7.1</a> )

Table 1-16. Parameters for SDK6 API Middleware CFS API **AmpCFS\_fclose()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-17. Returns for SDK6 Middleware CFS API **AmpCFS\_fclose()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.9 AmpCFS\_feof

### API Syntax:

**AmpCFS\_feof** ( AMP\_CFS\_FILE\_s \* file )

### Function Description:

- This function is used to check the end of a file.

### Parameters:

Type	Parameter	Description
AMP_CFS_FILE_s *	<b>file</b>	CFS file descriptor ( <a href="#">Section 1.2.7.1</a> )

Table 1-18. Parameters for SDK6 API Middleware CFS API **AmpCFS\_feof()**.

### Returns:

Return	Description
1	The EOF has been reached
0	The EOF has not been reached, or an internal error occurred
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-19. Returns for SDK6 Middleware CFS API **AmpCFS\_feof()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 1.2.10 AmpCFS\_FGetLen

### API Syntax:

**AmpCFS\_FGetLen** ( AMP\_CFS\_FILE\_s \* file )

### Function Description:

- This function is used to get the length of a file.

### Parameters:

Type	Parameter	Description
AMP_CFS_FILE_s *	<b>file</b>	File descriptor ( <a href="#">Section 1.2.7.1</a> )

Table 1-20. Parameters for SDK6 API Middleware CFS API **AmpCFS\_FGetLen()**.

### Returns:

Return	Description
>= 0	File length
-1	Failure

Table 1-21. Returns for SDK6 Middleware CFS API **AmpCFS\_FGetLen()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.11 AmpCFS\_FirstDirEnt

### API Syntax:

**AmpCFS\_FirstDirEnt** ( const char \* dirName, unsigned char attribute, AMP\_CFS\_DTA \* dirEntry )

### Function Description:

- This function is used to search files and return the first matched result.

### Parameters:

Type	Parameter	Description
const char *	<b>dirName</b>	The directory path being searched
unsigned char	<b>attribute</b>	Directory attributes
AMP_CFS_DTA *	<b>dirEntry</b>	The returned directory entry

Table 1-22. Parameters for SDK6 API Middleware CFS API **AmpCFS\_FirstDirEnt()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-23. Returns for SDK6 Middleware CFS API **AmpCFS\_FirstDirEnt()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.12 AmpCFS\_fopen

### API Syntax:

**AmpCFS\_fopen** ( AMP\_CFS\_FILE\_PARAM\_s \* fileParam)

### Function Description:

- This function is used to open a CFS file.

### Parameters:

Type	Parameter	Description
AMP_CFS_FILE_PARAM_s *	<b>fileParam</b>	Parameters used to open a file descriptor ( <a href="#">Section 1.2.12.1</a> )

Table 1-24. Parameters for SDK6 API Middleware CFS API **AmpCFS\_fopen()**.

### Returns:

Return	Description
File descriptor	A file descriptor keeps information for file operations such as file reading and writing.

Table 1-25. Returns for SDK6 Middleware CFS API **AmpCFS\_fopen()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 1.2.12.1 AmpCFS\_fopen > AMP\_CFS\_FILE\_PARAM\_s

Type	Field	Description
char [MAX_FILE-NAME_LENGTH]	<b>Filename</b>	File name
AMP_CFS_FILE_PARAM_ASYNC_s	<b>AsyncData</b>	Parameters related to asynchronous mode ( <a href="#">Section 1.2.12.2</a> )
UINT32	<b>Alignment</b>	File alignment: (The value should be a multiple of a cluster size)
UINT32	<b>BytesToSync</b>	The number of bytes that the CFS stream would invoke fsync() one time (A file should be synced after <b>BytesToSync</b> has been written)
UINT8	<b>Mode</b>	File open mode (See AMP_CFS_FILE_MODE_e)
BOOL8	<b>AsyncMode</b>	The flag to enable async mode (indicates whether a stream is in asynchronous mode)
BOOL8	<b>LowPriority</b>	The flag to enable low priority (indicates whether a stream runs in low priority)

Table 1-26. Definition of **AMP\_CFS\_FILE\_PARAM\_s** for CFS API **AmpCFS\_fopen()**.

### 1.2.12.2 AmpCFS\_fopen > AMP\_CFS\_FILE\_PARAM\_ASYNC\_s

Type	Field	Description
UINT8	<b>MaxNumBank</b>	Maximum number of banks used by a stream

Table 1-27. Definition of **AMP\_CFS\_FILE\_PARAM\_ASYNC\_s** for CFS API **AmpCFS\_fopen()**.

## 1.2.13 AmpCFS\_Format

### API Syntax:

**AmpCFS\_Format** (char driveName, const char \* param)

### Function Description:

- This function is used to format a drive.

### Parameters:

Type	Parameter	Description
char	<b>driveName</b>	Drive name (from A to Z)
const char *	<b>param</b>	The parameter string used to notify the driver of the format type

Table 1-28. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Format()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-29. Returns for SDK6 Middleware CFS API **AmpCFS\_Format()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.14 AmpCFS\_fread

### API Syntax:

**AmpCFS\_fread** ( void \* buffer, UINT64 size, UINT64 count, AMP\_CFS\_FILE\_s \* file )

### Function Description:

- This function is used to read data elements from a file, and stores them in a buffer.

### Parameters:

Type	Parameter	Description
void *	<b>buffer</b>	The buffer in which data elements are stored
UINT64	<b>size</b>	The size of each data element being read
UINT64	<b>count</b>	The number of data elements
AMP_CFS_FILE_s *	<b>file</b>	File descriptor ( <a href="#">Section 1.2.7.1</a> )

Table 1-30. Parameters for SDK6 API Middleware CFS API **AmpCFS\_fread()**.

### Returns:

Return	Description
Number of elements read	Number of elements read

Table 1-31. Returns for SDK6 Middleware CFS API **AmpCFS\_fread()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.15 AmpCFS\_fseek

### API Syntax:

**AmpCFS\_fseek** (AMP\_CFS\_FILE\_s \* file, INT64 offset, int origin)

### Function Description:

- This function is used to move the file I/O pointer.

### Parameters:

Type	Parameter	Description
AMP_CFS_FILE_s *	<b>file</b>	File descriptor ( <a href="#">Section 1.2.7.1</a> )
INT64	<b>offset</b>	Number of bytes to offset from the origin position
int	<b>origin</b>	Position used as a reference for the offset

Table 1-32. Parameters for SDK6 API Middleware CFS API **AmpCFS\_fseek()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-33. Returns for SDK6 Middleware CFS API **AmpCFS\_fseek()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.16 AmpCFS\_FStatus

### API Syntax:

**AmpCFS\_FStatus** ( const char \* fileName )

### Function Description:

- This function is used to get file or directory information.

### Parameters:

Type	Parameter	Description
const char *	<b>fileName</b>	The file path used to get the file status

Table 1-34. Parameters for SDK6 API Middleware CFS API **AmpCFS\_FStatus()**.

### Returns:

Return	Description
AMP_CFS_STATUS_UN-USED	The status indicating that a file is unused
AMP_CFS_STATUS_OPENED_READ	The status indicating that a file is opened to read
AMP_CFS_STATUS_OPENED_WRITE	The status indicating that a file is opened to write
AMP_CFS_STATUS_CLOSING_READ	The status indicating that a file is closing (async read mode only)
AMP_CFS_STATUS_CLOSING_WRITE	The status indicating that a file is closing (async write mode only)
AMP_CFS_STATUS_ERROR	The status indicating that an error occurs in operating a file (to block subsequent read/write)

Table 1-35. Returns for SDK6 Middleware CFS API **AmpCFS\_FStatus()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 1.2.17 AmpCFS\_FSync

### API Syntax:

**AmpCFS\_FSync** (AMP\_CFS\_FILE\_s \* file)

### Function Description:

- This function is used to flush all data in the cache for the specified file to media.

### Parameters:

Type	Parameter	Description
AMP_CFS_FILE_s *	<b>file</b>	File descriptor ( <a href="#">Section 1.2.7.1</a> )

Table 1-36. Parameters for SDK6 API Middleware CFS API **AmpCFS\_FSync()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-37. Returns for SDK6 Middleware CFS API **AmpCFS\_FSync()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.18 AmpCFS\_ftell

### API Syntax:

**AmpCFS\_ftell** (AMP\_CFS\_FILE\_s \* file)

### Function Description:

- This function is used to get the current file I/O pointer.

### Parameters:

Type	Parameter	Description
AMP_CFS_FILE_s *	<b>file</b>	File descriptor ( <a href="#">Section 1.2.7.1</a> )

Table 1-38. Parameters for SDK6 API Middleware CFS API **AmpCFS\_ftell()**.

### Returns:

Return	Description
File position	Current position of a file

Table 1-39. Returns for SDK6 Middleware CFS API **AmpCFS\_ftell()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.19 AmpCFS\_fwrite

### API Syntax:

**AmpCFS\_fwrite** ( const void \* buffer, UINT64 size, UINT64 count, AMP\_CFS\_FILE\_s \* file )

### Function Description:

- This function is used to write data elements from a buffer to a file.

### Parameters:

Type	Parameter	Description
const void *	<b>buffer</b>	The buffer from which data elements are read
UINT64	<b>size</b>	The size of each data element being written
UINT64	<b>count</b>	The number of data elements
AMP_CFS_FILE_s *	<b>file</b>	File descriptor ( <a href="#">Section 1.2.7.1</a> )

Table 1-40. Parameters for SDK6 API Middleware CFS API **AmpCFS\_fwrite()**.

### Returns:

Return	Description
Number of elements written	Number of elements written

Table 1-41. Returns for SDK6 Middleware CFS API **AmpCFS\_fwrite()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.20 AmpCFS\_GetCachedDataSize

### API Syntax:

**AmpCFS\_GetCachedDataSize** ( char driveName, UINT64 \* sizeByte )

### Function Description:

- This function is used to get the cached data size of a drive.

### Parameters:

Type	Parameter	Description
char	<b>driveName</b>	Drive name (from A to Z)
UINT64 *	<b>sizeByte</b>	The returned size of cached data

Table 1-42. Parameters for SDK6 API Middleware CFS API **AmpCFS\_GetCachedDataSize()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-43. Returns for SDK6 Middleware CFS API **AmpCFS\_GetCachedDataSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.21 AmpCFS\_GetDefaultCfg

### API Syntax:

**AmpCFS\_GetDefaultCfg** (AMP\_CFS\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the CFS module.

### Parameters:

Type	Parameter	Description
AMP_CFS_CFG_s *	<b>config</b>	The returned configuration of the CFS module ( <a href="#">Section 1.2.21.1</a> )

Table 1-44. Parameters for SDK6 API Middleware CFS API **AmpCFS\_GetDefaultCfg()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-45. Returns for SDK6 Middleware CFS API **AmpCFS\_GetDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 1.2.21.1 AmpCFS\_GetDefaultCfg > AMP\_CFS\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The working buffer of the CFS module
UINT32	<b>BufferSize</b>	The size of the working buffer
int (*)(int, UINT32)	<b>FileOperation</b>	The callback function to report all events of file operations in the CFS module
UINT32	<b>CacheMaxFileNum</b>	Maximum number of cached files
UINT32	<b>SchBankSize</b>	Size of a bank
AMP_TASK_INFO_s	<b>TaskInfo</b>	The information of CFS background task ( <a href="#">Section 2.2.13.2</a> )
UINT32 [AMP_CFS_MAX_TASK_AMOUNT]	<b>SchLowTxRate</b>	Minimum acceptable transmission rate Unit: KBps; it is for AsyncMode
UINT32 [AMP_CFS_MAX_TASK_AMOUNT]	<b>SchLowSpeedSize</b>	Tolerance size of writing data continually in low speed
UINT8 [AMP_CFS_MAX_DRIVE_AMOUNT]	<b>SchTaskDriveTable</b>	Mapping table used to convert drive name to task ID
BOOL8	<b>CacheEnable</b>	The flag used to enable or disable the cache function of the CFS module (if the value is TRUE, the cache function will be enabled)
UINT8	<b>SchTaskAmount</b>	Number of tasks
UINT8	<b>SchBankAmount</b>	Number of banks

Table 1-46. Definition of AMP\_CFS\_CFG\_s for CFS API AmpCFS\_GetDefaultCfg().

## 1.2.22 AmpCFS\_GetDev

### API Syntax:

**AmpCFS\_GetDev** (char driveName, AMP\_CFS\_DEVINF \* devInfo)

### Function Description:

- This function is used to get device capacity (i.e., the information of a drive).

### Parameters:

Type	Parameter	Description
char	<b>driveName</b>	Drive name (from A to Z)
AMP_CFS_DEVINF *	<b>devInfo</b>	Returned information of device capacity

Table 1-47. Parameters for SDK6 API Middleware CFS API **AmpCFS\_GetDev()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-48. Returns for SDK6 Middleware CFS API **AmpCFS\_GetDev()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.23 AmpCFS\_GetError

### API Syntax:

**AmpCFS\_GetError** ( int \* errNum)

### Function Description:

- This function is used to get the last error number.

### Parameters:

Type	Parameter	Description
int *	<b>errNum</b>	The returned error number

Table 1-49. Parameters for SDK6 API Middleware CFS API **AmpCFS\_GetError()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-50. Returns for SDK6 Middleware CFS API **AmpCFS\_GetError()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 1.2.24 AmpCFS\_GetFileMaxCachedDataSize

### API Syntax:

**AmpCFS\_GetFileMaxCachedDataSize** (AMP\_CFS\_FILE\_s \*file, UINT64 \*dataSize)

### Function Description:

- This function is used to get the max cached data size of a file.

### Parameters:

Type	Parameter	Description
AMP_CFS_FILE_s *	<b>file</b>	The file descriptor
UINT64*	<b>dataSize</b>	The returned data size (bytes)

Table 1-51. Parameters for SDK6 API Middleware CFS API **AmpCFS\_GetFileMaxCachedDataSize()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-52. Returns for SDK6 Middleware CFS API **AmpCFS\_GetFileMaxCachedDataSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.25 AmpCFS\_GetFileParam

### API Syntax:

**AmpCFS\_GetFileParam** ( AMP\_CFS\_FILE\_PARAM\_s \* fileParam)

### Function Description:

- This function is used to get default file parameters for opening a file descriptor.

### Parameters:

Type	Parameter	Description
AMP_CFS_FILE_PARAM_s *	<b>fileParam</b>	The returned file parameters ( <a href="#">Section 1.2.12.1</a> )

Table 1-53. Parameters for SDK6 API Middleware CFS API **AmpCFS\_GetFileParam()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-54. Returns for SDK6 Middleware CFS API **AmpCFS\_GetFileParam()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.26 AmpCFS\_GetRequiredBufferSize

### API Syntax:

**AmpCFS\_GetRequiredBufferSize** ( UINT32 schBankSize, UINT8 schBankAmount, UINT32 schStackSize, UINT8 schTaskAmount, BOOL8 cacheEnable, UINT32 cacheMaxFileNum )

### Function Description:

- This function is used to get the required buffer size for initializing the CFS module.

### Parameters:

Type	Parameter	Description
UINT32	<b>schBankSize</b>	Bank size
UINT8	<b>schBankAmount</b>	Number of banks
UINT32	<b>schStackSize</b>	The stack size of scheduling tasks
UINT8	<b>schTaskAmount</b>	The number of scheduling tasks
BOOL8	<b>cacheEnable</b>	Indicate if the cache is enabled
UINT32	<b>cacheMaxFileNum</b>	Maximum number of cached files

Table 1-55. Parameters for SDK6 API Middleware CFS API **AmpCFS\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 1-56. Returns for SDK6 Middleware CFS API **AmpCFS\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.27 AmpCFS\_Init

### API Syntax:

**AmpCFS\_Init** (AMP\_CFS\_CFG\_s \* config)

### Function Description:

- This function is used to initialize the CFS module.

### Parameters:

Type	Parameter	Description
AMP_CFS_CFG_s *	<b>config</b>	The configuration used to initialize the CFS module ( <a href="#">Section 1.2.21.1</a> )

Table 1-57. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Init()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-58. Returns for SDK6 Middleware CFS API **AmpCFS\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.28 AmpCFS\_Mkdir

### API Syntax:

**AmpCFS\_Mkdir** (const char \* dirName)

### Function Description:

- This function is used to create a new directory.

### Parameters:

Type	Parameter	Description
const char *	<b>dirName</b>	The path of a directory being created

Table 1-59. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Mkdir()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-60. Returns for SDK6 Middleware CFS API **AmpCFS\_Mkdir()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.29 AmpCFS\_Mount

### API Syntax:

**AmpCFS\_Mount** (char driveName)

### Function Description:

- This function is used to mount a drive.

### Parameters:

Type	Parameter	Description
char	<b>driveName</b>	Drive name (from A to Z)

Table 1-61. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Mount()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-62. Returns for SDK6 Middleware CFS API **AmpCFS\_Mount()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.30 AmpCFS\_Move

### API Syntax:

**AmpCFS\_Move** (const char \* srcFileName, const char \* dstFileName)

### Function Description:

- This function is used to move a source file to a destination location.

### Parameters:

Type	Parameter	Description
const char *	<b>srcFileName</b>	The path of a source file
const char *	<b>dstFileName</b>	The path of a destination file

Table 1-63. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Move()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-64. Returns for SDK6 Middleware CFS API **AmpCFS\_Move()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.31 AmpCFS\_NextDirEnt

### API Syntax:

**AmpCFS\_NextDirEnt** ( AMP\_CFS\_DTA \* dirEntry )

### Function Description:

- This function is used to search files and return the next matched result.

### Parameters:

Type	Parameter	Description
AMP_CFS_DTA *	<b>dirEntry</b>	The returned directory entry

Table 1-65. Parameters for SDK6 API Middleware CFS API **AmpCFS\_NextDirEnt()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-66. Returns for SDK6 Middleware CFS API **AmpCFS\_NextDirEnt()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 1.2.32 AmpCFS\_remove

### API Syntax:

**AmpCFS\_remove** (const char \* fileName)

### Function Description:

- This function is used to delete a file.

### Parameters:

Type	Parameter	Description
const char *	<b>fileName</b>	The path of a file being deleted

Table 1-67. Parameters for SDK6 API Middleware CFS API **AmpCFS\_remove()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-68. Returns for SDK6 Middleware CFS API **AmpCFS\_remove()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 1.2.33 AmpCFS\_Rmdir

#### API Syntax:

**AmpCFS\_Rmdir** (const char \* dirName)

#### Function Description:

- This function is used to delete a directory.

#### Parameters:

Type	Parameter	Description
const char *	<b>dirName</b>	The path of a directory being deleted

Table 1-69. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Rmdir()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-70. Returns for SDK6 Middleware CFS API **AmpCFS\_Rmdir()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.34 AmpCFS\_Stat

### API Syntax:

**AmpCFS\_Stat** (const char \* fileName, AMP\_CFS\_STAT \* status)

### Function Description:

- This function is used to get the status of a file.

### Parameters:

Type	Parameter	Description
const char *	<b>fileName</b>	The file path used to obtain its state in the file system
AMP_CFS_STAT *	<b>status</b>	The returned state of the file

Table 1-71. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Stat()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-72. Returns for SDK6 Middleware CFS API **AmpCFS\_Stat()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.35 AmpCFS\_Sync

### API Syntax:

**AmpCFS\_Sync** ( char driveName, int mode )

### Function Description:

- This function is used to write all cached data for a specified drive back to the underlying device.

### Parameters:

Type	Parameter	Description
char	<b>driveName</b>	Drive name (from A to Z)
int	<b>mode</b>	Invalidation mode (in AMBA_FS_NINVALIDATE mode, caches will not be invalidated; however, in AMBA_FS_INVALIDATE mode, caches will be invalidated)

Table 1-73. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Sync()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-74. Returns for SDK6 Middleware CFS API **AmpCFS\_Sync()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 1.2.36 AmpCFS\_Unmount

### API Syntax:

**AmpCFS\_Unmount** ( char driveName)

### Function Description:

- This function is used to unmount a drive.

### Parameters:

Type	Parameter	Description
char	<b>driveName</b>	Drive name (from A to Z)

Table 1-75. Parameters for SDK6 API Middleware CFS API **AmpCFS\_Unmount()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 1-76. Returns for SDK6 Middleware CFS API **AmpCFS\_Unmount()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

# 2 Design Rule for Camera File System

## 2.1 DCF: Overview

This chapter provides information regarding the implementation of the DCF module and the APIs to browse/update the mapping of media objects to the files in storage.

## 2.2 DCF: List of Functions

- (Section 2.2.1) AmpDCF\_AddDirectory
- (Section 2.2.2) AmpDCF\_AddFile
- (Section 2.2.3) AmpDCF\_AddRoot
- (Section 2.2.4) AmpDCF\_CheckIdValid
- (Section 2.2.5) AmpDCF\_Create
- (Section 2.2.6) AmpDCF\_Delete
- (Section 2.2.7) AmpDCF\_DumpITM
- (Section 2.2.8) AmpDCF\_GetDefaultCfg
- (Section 2.2.9) AmpDCF\_GetDirectoryList
- (Section 2.2.10) AmpDCF\_GetFileList
- (Section 2.2.11) AmpDCF\_GetFirstDnum
- (Section 2.2.12) AmpDCF\_GetFirstId
- (Section 2.2.13) AmpDCF\_GetInitDefaultCfg
- (Section 2.2.14) AmpDCF\_GetLastDnum
- (Section 2.2.15) AmpDCF\_GetLastId
- (Section 2.2.16) AmpDCF\_GetNextDnum
- (Section 2.2.17) AmpDCF\_GetNextId
- (Section 2.2.18) AmpDCF\_GetPrevDnum
- (Section 2.2.19) AmpDCF\_GetPrevId
- (Section 2.2.20) AmpDCF\_GetRequiredBufferSize
- (Section 2.2.21) AmpDCF\_Init
- (Section 2.2.22) AmpDCF\_RelDirectoryList
- (Section 2.2.23) AmpDCF\_RelFileList
- (Section 2.2.24) AmpDCF\_RemoveDirectory
- (Section 2.2.25) AmpDCF\_RemoveFile
- (Section 2.2.26) AmpDCF\_RemoveRoot
- (Section 2.2.27) AmpDCF\_Scan

## 2.2.1 AmpDCF\_AddDirectory

### API Syntax:

**AmpDCF\_AddDirectory** ( AMP\_DCF\_HDLR\_s \* hdlr, char \* name )

### Function Description:

- This function is used to add a directory to a DCF handler.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	<b>hdlr</b>	The DCF handler ( <a href="#">Section 2.2.1.1</a> )
char *	<b>name</b>	The directory name

Table 2-1. Parameters for SDK6 API Middleware DCF API **AmpDCF\_AddDirectory()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-2. Returns for SDK6 Middleware DCF API **AmpDCF\_AddDirectory()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 2.2.1.1 AmpDCF\_AddDirectory > AMP\_DCF\_HDLR\_s

Type	Field	Description
AMP_DCF_FILTER_s *	<b>Filter</b>	DCF name filter ( <a href="#">Section 2.2.1.2</a> )

Table 2-3. Definition of **AMP\_DCF\_HDLR\_s** for DCF API **AmpDCF\_AddDirectory()**.

### 2.2.1.2 AmpDCF\_AddDirectory > AMP\_DCF\_FILTER\_s

Type	Field	Description
UINT32 (*) (char *)	<b>NameToDnum</b>	The interface to convert name to directory number
UINT32 (*) (char *)	<b>NameToId</b>	The interface to convert name to ID
UINT32 (*) (UINT32, UINT32)	<b>GetId</b>	The interface to get an ID in accordance with a directory number and a file number
UINT32 (*) (UINT32)	<b>IdToDnum</b>	The interface to convert ID to directory number
UINT32 (*) (UINT32)	<b>IdToFnum</b>	The interface to convert ID to file number
int (*)(char *, AMP_DCF_TABLE_HDLR_s *)	<b>ScanDirectory</b>	The interface to scan directories in a specified root into a DCF table ( <a href="#">Section 2.2.1.3</a> )
int (*)(char *, AMP_DCF_TABLE_HDLR_s *)	<b>ScanFile</b>	The interface to scan files in a specified root into a DCF table ( <a href="#">Section 2.2.1.3</a> )

Table 2-4. Definition of **AMP\_DCF\_FILTER\_s** for DCF API **AmpDCF\_AddDirectory()**.

### 2.2.1.3 AmpDCF\_AddDirectory > AMP\_DCF\_TABLE\_HDLR\_s

Type	Field	Description
AMP_DCF_TABLE_s *	<b>Func</b>	Functions of a table handler ( <a href="#">Section 2.2.1.4</a> )

Table 2-5. Definition of **AMP\_DCF\_TABLE\_HDLR\_s** for DCF API **AmpDCF\_AddDirectory()**.

### 2.2.1.4 AmpDCF\_AddDirectory > AMP\_DCF\_TABLE\_s

Type	Field	Description
int (*)(UINT32, UINT32, AMP_DCF_TABLE_HDLR_s **)	<b>Create</b>	The interface to create a table handler
int (*)(AMP_DCF_TABLE_HDLR_s *)	<b>Delete</b>	The interface to delete a table handler
int (*)(AMP_DCF_TABLE_HDLR_s *, UINT32, char *)	<b>AddDirectory</b>	The interface to add a directory
int (*)(AMP_DCF_TABLE_HDLR_s *, char *)	<b>RemoveDirectory</b>	The interface to remove a directory (It could remove a parent directory, so the table must be scanned.)
int (*)(AMP_DCF_TABLE_HDLR_s *, UINT32, UINT32, char *)	<b>AddFile</b>	The interface to add a file



Type	Field	Description
int(*) (AMP_DCF_TABLE_HDLR_s *, UINT32, UINT32, char *)	<b>RemoveFile</b>	The interface to remove a file
int(*) (AMP_DCF_TABLE_HDLR_s *)	<b>GetFirstDnum</b>	The interface to get the first directory number (fnum would be set to first)
int(*) (AMP_DCF_TABLE_HDLR_s *)	<b>GetLastDnum</b>	The interface to get the last directory number (fnum would be set to last)
int(*) (AMP_DCF_TABLE_HDLR_s *, UINT32)	<b>GetNextDnum</b>	The interface to get the next directory number from the given dnum (dnum is just for reference, could be invalid) (fnum would be set to first)
int(*) (AMP_DCF_TABLE_HDLR_s *, UINT32)	<b>GetPrevDnum</b>	The interface to get the previous directory number from the given dnum (dnum is just for reference, could be invalid) (fnum would be set to last)
int(*) (AMP_DCF_TABLE_HDLR_s *, UINT32)	<b>GetFirstFnum</b>	The interface to get the first file number (hdlr, dnum)
int(*) (AMP_DCF_TABLE_HDLR_s *, UINT32)	<b>GetLastFnum</b>	The interface to get the last file number (hdlr, dnum)
int(*) (AMP_DCF_TABLE_HDLR_s *, UINT32, UINT32)	<b>GetNextFnum</b>	The interface to get the next file number from the given dnum and fnum (hdlr, dnum, fnum) (dnum and fnum are just for reference, could be invalid)
int(*) (AMP_DCF_TABLE_HDLR_s *, UINT32, UINT32)	<b>GetPrevFnum</b>	The interface to get the previous file number from the given dnum and fnum (hdlr, dnum, fnum) (dnum and fnum are just for reference, could be invalid)
int(*) (AMP_DCF_TABLE_HDLR_s *, UINT32, AMP_DCF_DIR_LIST_s *)	<b>GetDirectoryList</b>	The interface to get a list of directories with a specified number (hdlr, dnum, bytewidth) ( <a href="#">Section 2.2.1.5</a> )
int(*) (AMP_DCF_TABLE_HDLR_s *, UINT32, UINT32, AMP_DCF_FILE_LIST_s *)	<b>GetFileList</b>	The interface to get a list of files with a specified number (hdlr, fnum, bytewidth) ( <a href="#">Section 2.2.1.7</a> )
BOOL(*) (AMP_DCF_TABLE_HDLR_s *, UINT32, UINT32)	<b>CheckIdValid</b>	The interface to check if an input ID is already in table, which is combined from a file number and a directory number (hdlr, dnum, fnum)

Table 2-6. Definition of **AMP\_DCF\_TABLE\_s** for DCF API **AmpDCF\_AddDirectory()**.

### 2.2.1.5 AmpDCF\_AddDirectory > AMP\_DCF\_DIR\_LIST\_s

Type	Field	Description
UINT32	<b>Count</b>	Directory count
AMP_DCF_DIR_s *	<b>DirList</b>	Directory list ( <a href="#">Section 2.2.1.6</a> )

Table 2-7. Definition of **AMP\_DCF\_DIR\_LIST\_s** for DCF API **AmpDCF\_AddDirectory()**.

### 2.2.1.6 AmpDCF\_AddDirectory > AMP\_DCF\_DIR\_s

Type	Field	Description
char [MAX_FILE-NAME_LENGTH]	<b>Name</b>	Directory name

Table 2-8. Definition of **AMP\_DCF\_DIR\_s** for DCF API **AmpDCF\_AddDirectory()**.

### 2.2.1.7 AmpDCF\_AddDirectory > AMP\_DCF\_FILE\_LIST\_s

Type	Field	Description
UINT32	<b>Count</b>	File count
AMP_DCF_FILE_s *	<b>FileList</b>	File list ( <a href="#">Section 2.2.1.8</a> )

Table 2-9. Definition of **AMP\_DCF\_FILE\_LIST\_s** for DCF API **AmpDCF\_AddDirectory()**.

### 2.2.1.8 AmpDCF\_AddDirectory > AMP\_DCF\_FILE\_s

Type	Field	Description
char [MAX_FILE-NAME_LENGTH]	<b>Name</b>	File name

Table 2-10. Definition of **AMP\_DCF\_FILE\_s** for DCF API **AmpDCF\_AddDirectory()**.

## 2.2.2 AmpDCF\_AddFile

### API Syntax:

**AmpDCF\_AddFile** ( AMP\_DCF\_HDLR\_s \* hdlr, char \* name )

### Function Description:

- This function is used to add a file to a DCF handler.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	<b>hdlr</b>	The DCF handler ( <a href="#">Section 2.2.1.1</a> )
char *	<b>name</b>	The filename

Table 2-11. Parameters for SDK6 API Middleware DCF API **AmpDCF\_AddFile()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-12. Returns for SDK6 Middleware DCF API **AmpDCF\_AddFile()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.3 AmpDCF\_AddRoot

### API Syntax:

**AmpDCF\_AddRoot** ( AMP\_DCF\_HDLR\_s \* hdlr, char \* path )

### Function Description:

- This function is used to add a DCF root into a DCF handler. This only scans sub-directories into the DCF table.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	<b>hdlr</b>	The DCF handler ( <a href="#">Section 2.2.1.1</a> )
char *	<b>path</b>	The path of a DCF root being added (e.g., c:\DCIM)

Table 2-13. Parameters for SDK6 API Middleware DCF API **AmpDCF\_AddRoot()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-14. Returns for SDK6 Middleware DCF API **AmpDCF\_AddRoot()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.4 AmpDCF\_CheckIdValid

### API Syntax:

**AmpDCF\_CheckIdValid** ( AMP\_DCF\_HDLR\_s \* hdlr, UINT32 id )

### Function Description:

- This function is used to check whether an object is in the table.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	<b>hdlr</b>	The DCF handler ( <a href="#">Section 2.2.1.1</a> )
UINT32	<b>id</b>	Object ID

Table 2-15. Parameters for SDK6 API Middleware DCF API **AmpDCF\_CheckIdValid()**.

### Returns:

Return	Description
TRUE	Object is in the DCF table
FALSE	Object is not in the DCF table

Table 2-16. Returns for SDK6 Middleware DCF API **AmpDCF\_CheckIdValid()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.5 AmpDCF\_Create

### API Syntax:

**AmpDCF\_Create** ( AMP\_DCF\_CFG\_s \* config, AMP\_DCF\_HDLR\_s \*\* hdlr )

### Function Description:

- This function is used to create a DCF instance. Multiple DCF instances are supported.

### Parameters:

Type	Parameter	Description
AMP_DCF_CFG_s *	<b>config</b>	The default configuration for creating a DCF handler ( <a href="#">Section 2.2.5.1</a> )
AMP_DCF_HDLR_s **	<b>hdlr</b>	The returned DCF handler ( <a href="#">Section 2.2.1.1</a> )

Table 2-17. Parameters for SDK6 API Middleware DCF API **AmpDCF\_Create()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-18. Returns for SDK6 Middleware DCF API **AmpDCF\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 2.2.5.1 AmpDCF\_Create > AMP\_DCF\_CFG\_s

Type	Field	Description
AMP_DCF_FILTER_s *	<b>Filter</b>	DCF name filter ( <a href="#">Section 2.2.1.2</a> )
AMP_DCF_TABLE_s *	<b>Table</b>	DCF table ( <a href="#">Section 2.2.1.4</a> )
UINT32	<b>MaxDir</b>	The maximum number of directories in a DCF handler
UINT32	<b>MaxFile</b>	The maximum number of files in a DCF handler

Table 2-19. Definition of **AMP\_DCF\_CFG\_s** for DCF API **AmpDCF\_Create()**.

Confidential  
For HAOTEK Only

## 2.2.6 AmpDCF\_Delete

### API Syntax:

**AmpDCF\_Delete** ( AMP\_DCF\_HDLR\_s \* hdlr )

### Function Description:

- This function is used to delete a DCF handler.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	<b>hdlr</b>	The DCF handler being deleted ( <a href="#">Section 2.2.1.1</a> )

Table 2-20. Parameters for SDK6 API Middleware DCF API **AmpDCF\_Delete()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-21. Returns for SDK6 Middleware DCF API **AmpDCF\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 2.2.7 AmpDCF\_DumpITM

### API Syntax:

**AmpDCF\_DumpITM** (void )

### Function Description:

- This function is used to dump an ITM file.

### Parameters:

None

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-22. Returns for SDK6 Middleware DCF API **AmpDCF\_DumpITM()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.8 AmpDCF\_GetDefaultCfg

### API Syntax:

**AmpDCF\_GetDefaultCfg** ( AMP\_DCF\_CFG\_s \* config )

### Function Description:

- This function is used to get the default configuration for creating a DCF handler.

### Parameters:

Type	Parameter	Description
AMP_DCF_CFG_s *	<b>config</b>	The returned configuration of a DCF handler ( <a href="#">Section 2.2.5.1</a> )

Table 2-23. Parameters for SDK6 API Middleware DCF API **AmpDCF\_GetDefaultCfg()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-24. Returns for SDK6 Middleware DCF API **AmpDCF\_GetDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.9 AmpDCF\_GetDirectoryList

### API Syntax:

**AmpDCF\_GetDirectoryList** ( AMP\_DCF\_HDLR\_s \* hdlr, UINT32 dnum )

### Function Description:

- This function is used to get a list of the directories with the specified directory number. The list should be released via AmpDCF\_RelDirectoryList() once the list is no longer used.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	<b>hdlr</b>	The DCF handler ( <a href="#">Section 2.2.1.1</a> )
UINT32	<b>dnum</b>	The directory number

Table 2-25. Parameters for SDK6 API Middleware DCF API **AmpDCF\_GetDirectoryList()**.

### Returns:

Return	Description
The directory list	AMP_DCF_DIR_LIST_s * (Please refer to <a href="#">Section 2.2.1.5</a> )

Table 2-26. Returns for SDK6 Middleware DCF API **AmpDCF\_GetDirectoryList()**.

### Example:

Please refer to Unit Test document.

### Note:

The directory list must be released, once the list is no longer used.

## 2.2.10 AmpDCF\_GetFileList

### API Syntax:

**AmpDCF\_GetFileList** ( AMP\_DCF\_HDLR\_s \* hdlr, UINT32 id )

### Function Description:

- This function is used to get the list of files with a specified ID. The list should be released via AmpDCF\_RelFileList() once the list is no longer used.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	<b>hdlr</b>	The DCF handler ( <a href="#">Section 2.2.1.1</a> )
UINT32	<b>id</b>	The object ID

Table 2-27. Parameters for SDK6 API Middleware DCF API **AmpDCF\_GetFileList()**.

### Returns:

Return	Description
The file list	AMP_DCF_FILE_LIST_s * (Please refer to <a href="#">Section 2.2.1.7</a> )

Table 2-28. Returns for SDK6 Middleware DCF API **AmpDCF\_GetFileList()**.

### Example:

Please refer to Unit Test document.

### Note:

The file list must be released, once the list is no longer used.

## 2.2.11 AmpDCF\_GetFirstDnum

### API Syntax:

**AmpDCF\_GetFirstDnum** ( AMP\_DCF\_HDLR\_s \* hdlr )

### Function Description:

- This function is used to get the first directory number.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	hdlr	The DCF handler ( <a href="#">Section 2.2.1.1</a> )

Table 2-29. Parameters for SDK6 API Middleware DCF API **AmpDCF\_GetFirstDnum()**.

### Returns:

Return	Description
The first directory number	The first directory number

Table 2-30. Returns for SDK6 Middleware DCF API **AmpDCF\_GetFirstDnum()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.12 AmpDCF\_GetFirstId

### API Syntax:

**AmpDCF\_GetFirstId** ( AMP\_DCF\_HDLR\_s \* hdlr )

### Function Description:

- This function is used to get the first DCF object ID.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	hdlr	The DCF handler ( <a href="#">Section 2.2.1.1</a> )

Table 2-31. Parameters for SDK6 API Middleware DCF API **AmpDCF\_GetFirstId()**.

### Returns:

Return	Description
Object ID	The first DCF object ID

Table 2-32. Returns for SDK6 Middleware DCF API **AmpDCF\_GetFirstId()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.13 AmpDCF\_GetInitDefaultCfg

### API Syntax:

**AmpDCF\_GetInitDefaultCfg** ( AMP\_DCF\_INIT\_CFG\_s \* config )

### Function Description:

- This function is used to get the default configuration for initializing the DCF module.

### Parameters:

Type	Parameter	Description
AMP_DCF_INIT_CFG_s *	<b>config</b>	The returned configuration of the DCF module ( <a href="#">Section 2.2.13.1</a> )

Table 2-33. Parameters for SDK6 API Middleware DCF API **AmpDCF\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-34. Returns for SDK6 Middleware DCF API **AmpDCF\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 2.2.13.1 AmpDCF\_GetInitDefaultCfg > AMP\_DCF\_INIT\_CFG\_s

Type	Field	Description
AMP_TASK_INFO_s	<b>TaskInfo</b>	DCF task information ( <a href="#">Section 2.2.13.2</a> )
void *	<b>Buffer</b>	The work buffer of the DCF module
UINT32	<b>BufferSize</b>	The work buffer size of DCF module
UINT32	<b>MaxDirPerDnum</b>	The maximum number of directories per directory number
UINT32	<b>MaxFilePerId</b>	The maximum number of files per ID
UINT32	<b>MaxPendingOp</b>	The maximum pending operations
AMP_DCF_ITM_CFG_s	<b>ItmCfg</b>	ITM configuration ( <a href="#">Section 2.2.13.3</a> )
AMP_DCF_DEF_TBL_CFG_s	<b>DefTblCfg</b>	The configuration of the default table (This only works when EnableDefTbl is TRUE) ( <a href="#">Section 2.2.13.4</a> )
UINT8	<b>MaxHdlr</b>	The maximum number of DCF handlers in the DCF module
BOOL8	<b>EnableITM</b>	The flag used to enable the ITM function
BOOL8	<b>EnableDefTbl</b>	The flag used to enable the default table

Table 2-35. Definition of AMP\_DCF\_INIT\_CFG\_s for DCF API AmpDCF\_GetInitDefaultCfg().

### 2.2.13.2 AmpDCF\_GetInitDefaultCfg > AMP\_TASK\_INFO\_s

Type	Field	Description
UINT32	<b>Priority</b>	Task priority
UINT32	<b>StackSize</b>	Stack size of tasks
UINT32	<b>CoreSelection</b>	Core selection bitmap. Bit[0] = 1 means core #0 is selected.

Table 2-36. Definition of AMP\_TASK\_INFO\_s for DCF API AmpDCF\_GetInitDefaultCfg().

### 2.2.13.3 AmpDCF\_GetInitDefaultCfg > AMP\_DCF\_ITM\_CFG\_s

Type	Field	Description
char [MAX_FILE-NAME_LENGTH]	<b>Name</b>	Name of the ITM file

Table 2-37. Definition of AMP\_DCF\_ITM\_CFG\_s for DCF API AmpDCF\_GetInitDefaultCfg().

### 2.2.13.4 AmpDCF\_GetInitDefaultCfg > AMP\_DCF\_DEF\_TBL\_CFG\_s

Type	Field	Description
UINT32	<b>MaxDir</b>	Maximum number of directories that the default table can keep
UINT32	<b>MaxFile</b>	Maximum number of files that the default table can keep
UINT8	<b>MaxHdlr</b>	Maximum number of default table handlers

Table 2-38. Definition of AMP\_DCF\_DEF\_TBL\_CFG\_s for DCF API AmpDCF\_GetInitDefaultCfg().



## 2.2.14 AmpDCF\_GetLastDnum

### API Syntax:

**AmpDCF\_GetLastDnum** ( AMP\_DCF\_HDLR\_s \* hdlr )

### Function Description:

- This function is used to get the last directory number.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	hdlr	The DCF handler ( <a href="#">Section 2.2.1.1</a> )

Table 2-39. Parameters for SDK6 API Middleware DCF API **AmpDCF\_GetLastDnum()**.

### Returns:

Return	Description
Directory number	The last directory number

Table 2-40. Returns for SDK6 Middleware DCF API **AmpDCF\_GetLastDnum()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.15 AmpDCF\_GetLastId

### API Syntax:

**AmpDCF\_GetLastId** ( AMP\_DCF\_HDLR\_s \* hdlr )

### Function Description:

- This function is used to get the last DCF object ID.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	hdlr	The DCF handler ( <a href="#">Section 2.2.1.1</a> )

Table 2-41. Parameters for SDK6 API Middleware DCF API **AmpDCF\_GetLastId()**.

### Returns:

Return	Description
Object ID	The last DCF object ID

Table 2-42. Returns for SDK6 Middleware DCF API **AmpDCF\_GetLastId()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.16 AmpDCF\_GetNextDnum

### API Syntax:

**AmpDCF\_GetNextDnum** ( AMP\_DCF\_HDLR\_s \* hdlr )

### Function Description:

- This function is used to get the next nth directory number from the current directory (No Cycle).

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	hdlr	The DCF handler ( <a href="#">Section 2.2.1.1</a> )

Table 2-43. Parameters for SDK6 API Middleware DCF API **AmpDCF\_GetNextDnum()**.

### Returns:

Return	Description
Directory number	The next nth directory number

Table 2-44. Returns for SDK6 Middleware DCF API **AmpDCF\_GetNextDnum()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.17 AmpDCF\_GetNextId

### API Syntax:

**AmpDCF\_GetNextId** ( AMP\_DCF\_HDLR\_s \* hdlr )

### Function Description:

- This function is used to get the next nth object ID from the current ID (No Cycle).

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	hdlr	The DCF handler ( <a href="#">Section 2.2.1.1</a> )

Table 2-45. Parameters for SDK6 API Middleware DCF API **AmpDCF\_GetNextId()**.

### Returns:

Return	Description
Object ID	The next nth DCF object ID

Table 2-46. Returns for SDK6 Middleware DCF API **AmpDCF\_GetNextId()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.18 AmpDCF\_GetPrevDnum

### API Syntax:

**AmpDCF\_GetPrevDnum** ( AMP\_DCF\_HDLR\_s \* hdlr )

### Function Description:

- This function is used to get the previous nth directory number from the current directory (No Cycle).

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	hdlr	The DCF handler ( <a href="#">Section 2.2.1.1</a> )

Table 2-47. Parameters for SDK6 API Middleware DCF API **AmpDCF\_GetPrevDnum()**.

### Returns:

Return	Description
Directory number	The previous nth directory number

Table 2-48. Returns for SDK6 Middleware DCF API **AmpDCF\_GetPrevDnum()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.19 AmpDCF\_GetPrevId

### API Syntax:

**AmpDCF\_GetPrevId** ( AMP\_DCF\_HDLR\_s \* hdlr )

### Function Description:

- This function is used to get the previous nth object ID from the current ID (No Cycle).

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	hdlr	The DCF handler ( <a href="#">Section 2.2.1.1</a> )

Table 2-49. Parameters for SDK6 API Middleware DCF API **AmpDCF\_GetPrevId()**.

### Returns:

Return	Description
Object ID	The previous nth DCF object ID

Table 2-50. Returns for SDK6 Middleware DCF API **AmpDCF\_GetPrevId()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.20 AmpDCF\_GetRequiredBufferSize

### API Syntax:

**AmpDCF\_GetRequiredBufferSize** (UINT8 maxHdlr, UINT32 stackSize, UINT32 maxDirPerDnum, UINT32 maxFilePerId, UINT32 maxPendingOp, BOOL8 enableDefTable, UINT8 maxTblHdlr, UINT32 maxTblDir, UINT32 maxTblFile)

### Function Description:

- This function is used to get the required buffer size for initializing the DCF module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of DCF handlers
UINT32	<b>stackSize</b>	Stack size
UINT32	<b>maxDirPerDnum</b>	The maximum number of directories with the same number
UINT32	<b>maxFilePerId</b>	The maximum number of files with the same ID
UINT32	<b>maxPendingOp</b>	The maximum number of pending operations
BOOL8	<b>enableDefTable</b>	The flag to enable default DCF table (need some more memory space)
UINT8	<b>maxTblHdlr</b>	The maximum number of DCF table handlers (The value only works when EnableDefTable is TRUE.)
UINT32	<b>maxTblDir</b>	The maximum number of directories in the DCF module (The number is shared between all DCF tables, and only works when EnableDefTable is TRUE.)
UINT32	<b>maxTblFile</b>	The maximum number of files in the DCF module (The number is shared between all DCF tables, and only works when EnableDefTable is TRUE.)

Table 2-51. Parameters for SDK6 API Middleware DCF API **AmpDCF\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 2-52. Returns for SDK6 Middleware DCF API **AmpDCF\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.21 AmpDCF\_Init

### API Syntax:

**AmpDCF\_Init** ( AMP\_DCF\_INIT\_CFG\_s \* config )

### Function Description:

- This function is used to initialize the DCF module.

### Parameters:

Type	Parameter	Description
AMP_DCF_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the DCF module ( <a href="#">Section 2.2.13.1</a> )

Table 2-53. Parameters for SDK6 API Middleware DCF API **AmpDCF\_Init()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-54. Returns for SDK6 Middleware DCF API **AmpDCF\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 2.2.22 AmpDCF\_RelDirectoryList

### API Syntax:

**AmpDCF\_RelDirectoryList** (AMP\_DCF\_HDLR\_s \* hdlr, AMP\_DCF\_DIR\_LIST\_s \* list)

### Function Description:

- This function is used to release a directory list.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	<b>hdlr</b>	The DCF handler ( <a href="#">Section 2.2.1.1</a> )
AMP_DCF_DIR_LIST_s *	<b>list</b>	The directory list being released ( <a href="#">Section 2.2.1.5</a> )

Table 2-55. Parameters for SDK6 API Middleware DCF API **AmpDCF\_RelDirectoryList()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-56. Returns for SDK6 Middleware DCF API **AmpDCF\_RelDirectoryList()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.23 AmpDCF\_RelFileList

### API Syntax:

**AmpDCF\_RelFileList** (AMP\_DCF\_HDLR\_s \* hdlr, AMP\_DCF\_FILE\_LIST\_s \* list)

### Function Description:

- This function is used to release a file list.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	<b>hdlr</b>	The DCF handler ( <a href="#">Section 2.2.1.1</a> )
AMP_DCF_FILE_LIST_s *	<b>list</b>	The file list being released ( <a href="#">Section 2.2.1.7</a> )

Table 2-57. Parameters for SDK6 API Middleware DCF API **AmpDCF\_RelFileList()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-58. Returns for SDK6 Middleware DCF API **AmpDCF\_RelFileList()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.24 AmpDCF\_RemoveDirectory

### API Syntax:

**AmpDCF\_RemoveDirectory** (AMP\_DCF\_HDLR\_s \* hdlr, char \* name)

### Function Description:

- This function is used to remove a directory from a DCF handler.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	<b>hdlr</b>	The DCF handler ( <a href="#">Section 2.2.1.1</a> )
char *	<b>name</b>	The directory name

Table 2-59. Parameters for SDK6 API Middleware DCF API **AmpDCF\_RemoveDirectory()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-60. Returns for SDK6 Middleware DCF API **AmpDCF\_RemoveDirectory()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.25 AmpDCF\_RemoveFile

### API Syntax:

**AmpDCF\_RemoveFile** (AMP\_DCF\_HDLR\_s \* hdlr, char \* name)

### Function Description:

- This function is used to remove a file from a DCF handler.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	<b>hdlr</b>	The DCF handler ( <a href="#">Section 2.2.1.1</a> )
char *	<b>name</b>	The file name

Table 2-61. Parameters for SDK6 API Middleware DCF API **AmpDCF\_RemoveFile()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-62. Returns for SDK6 Middleware DCF API **AmpDCF\_RemoveFile()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.26 AmpDCF\_RemoveRoot

### API Syntax:

**AmpDCF\_RemoveRoot** (AMP\_DCF\_HDLR\_s \* hdlr, char \* path)

### Function Description:

- This function is used to remove a DCF root from a DCF handler.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	<b>hdlr</b>	The DCF handler ( <a href="#">Section 2.2.1.1</a> )
char *	<b>path</b>	The path of a DCF root being removed (e.g., c:\DCIM)

Table 2-63. Parameters for SDK6 API Middleware DCF API **AmpDCF\_RemoveRoot()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-64. Returns for SDK6 Middleware DCF API **AmpDCF\_RemoveRoot()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 2.2.27 AmpDCF\_Scan

### API Syntax:

**AmpDCF\_Scan** (AMP\_DCF\_HDLR\_s \* hdlr, BOOL async)

### Function Description:

- This function is used to scan all valid files into a DCF table.

### Parameters:

Type	Parameter	Description
AMP_DCF_HDLR_s *	<b>hdlr</b>	The DCF handler ( <a href="#">Section 2.2.1.1</a> )
BOOL	<b>async</b>	Async mode (In sync mode, all directories are scanned after the function return. In async mode, only the last directory is scanned after the return.)

Table 2-65. Parameters for SDK6 API Middleware DCF API **AmpDCF\_Scan()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 2-66. Returns for SDK6 Middleware DCF API **AmpDCF\_Scan()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

# 3 Demuxer

## 3.1 Demuxer: Overview

This chapter provides information regarding the Demuxer module. The primary function of the Demuxer module is to handle Demuxer pipes. Each Demuxer pipe includes multiple formats and their media information objects. Demuxer module will process the Demuxer pipes which are added to it.

The Demuxer module performs the following functions:

1. Initialize the Demuxer module
2. Create a Demuxer pipe
3. Delete a Demuxer pipe
4. Add a Demuxer pipe to Demuxer
5. Remove a Demuxer pipe from Demuxer
6. Other Demuxer-related functions

## 3.2 Demuxer: List of Functions

- [\(Section 3.2.1\) AmpDemuxer\\_Add](#)
- [\(Section 3.2.2\) AmpDemuxer\\_Create](#)
- [\(Section 3.2.3\) AmpDemuxer\\_Delete](#)
- [\(Section 3.2.4\) AmpDemuxer\\_FeedFrame](#)
- [\(Section 3.2.5\) AmpDemuxer\\_GetDefaultCfg](#)
- [\(Section 3.2.6\) AmpDemuxer\\_GetDefaultImageInfoCfg](#)
- [\(Section 3.2.7\) AmpDemuxer\\_GetDefaultMovieInfoCfg](#)
- [\(Section 3.2.8\) AmpDemuxer\\_GetDefaultSoundInfoCfg](#)
- [\(Section 3.2.9\) AmpDemuxer\\_GetInitDefaultCfg](#)
- [\(Section 3.2.10\) AmpDemuxer\\_GetRequiredBufferSize](#)
- [\(Section 3.2.11\) AmpDemuxer\\_Init](#)
- [\(Section 3.2.12\) AmpDemuxer\\_InitImageInfo](#)
- [\(Section 3.2.13\) AmpDemuxer\\_InitMovieInfo](#)
- [\(Section 3.2.14\) AmpDemuxer\\_InitSoundInfo](#)
- [\(Section 3.2.15\) AmpDemuxer\\_OnDataRequest](#)
- [\(Section 3.2.16\) AmpDemuxer\\_Remove](#)
- [\(Section 3.2.17\) AmpDemuxer\\_Seek](#)

- (Section 3.2.18) `AmpDemuxer_SetProcParam`
- (Section 3.2.19) `AmpDemuxer_Start`
- (Section 3.2.20) `AmpDemuxer_Stop`
- (Section 3.2.21) `AmpDemuxer_WaitComplete`

Confidential  
For HAOTEK Only



### 3.2.1 AmpDemuxer\_Add

#### API Syntax:

**AmpDemuxer\_Add** (AMP\_DEMUXER\_PIPE\_HDLR\_s \* pipe)

#### Function Description:

- This function is used to add a Demuxer pipe to the Demuxer module.

#### Parameters:

Type	Parameter	Description
AMP_DEMUXER_PIPE_HDLR_s *	<b>pipe</b>	The Demuxer pipe being added ( <a href="#">Section 3.2.1.1</a> )

Table 3-1. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_Add()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-2. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_Add()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

#### 3.2.1.1 AmpDemuxer\_Add > AMP\_DEMUXER\_PIPE\_HDLR\_s

Type	Field	Description
AMP_DMx_FORMAT_HDLR_s * [AMP_DEMUXER_MAX_FORMAT_PER_PIPE]	<b>Format</b>	Format handlers in a Demuxer pipe (AMP_DMx_FORMAT_HDLR_s is defined in <code>Format.h</code> , please refer to <a href="#">Section 5.4.1.2</a> )
UINT8	<b>FormatCount</b>	Number of Format handlers in a Demuxer pipe

Table 3-3. Definition of **AMP\_DEMUXER\_PIPE\_HDLR\_s** for Demuxer API **AmpDemuxer\_Add()**.

### 3.2.2 AmpDemuxer\_Create

#### API Syntax:

**AmpDemuxer\_Create** ( AMP\_DEMUXER\_PIPE\_CFG\_s \* config, AMP\_DEMUXER\_PIPE\_HDLR\_s \*\* pipe )

#### Function Description:

- This function is used to create a Demuxer pipe.

#### Parameters:

Type	Parameter	Description
AMP_DEMUXER_PIPE_CFG_s *	<b>config</b>	The configuration used to create a Demuxer pipe ( <a href="#">Section 3.2.2.1</a> )
AMP_DEMUXER_PIPE_HDLR_s **	<b>pipe</b>	The created pipe ( <a href="#">Section 3.2.1.1</a> )

Table 3-4. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_Create()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-5. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_Create()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.2.1 AmpDemuxer\_Create > AMP\_DEMUXER\_PIPE\_CFG\_s

Type	Field	Description
AMP_DMUX_FORMAT_HDLR_s * [AMP_DEMUXER_MAX_FORMAT_PER_PIPE]	<b>Format</b>	Format handlers in a pipe ( <a href="#">Section 5.4.1.2</a> )
AMP_MEDIA_INFO_s * [AMP_DEMUXER_MAX_FORMAT_PER_PIPE]	<b>Media</b>	Media information objects in a pipe ( <a href="#">Section 5.4.1.4</a> )
UINT32	<b>TaskPriority</b>	The task priority of a Demuxer pipe if TaskMode is AMP_DEMUXER_TASK_MODE_STANDALONE (The default value is the same as that of the Demuxer module.)
AMP_CALLBACK_f	<b>OnEvent</b>	The callback function for handling Demuxer events (AMP_CALLBACK_f is defined in <code>common.h</code> )
UINT32	<b>ProcParam</b>	The process parameters of a Demuxer pipe (In demuxing a movie and sound, the value means process duration (ms). In demuxing an image, the value means a frame number.)
UINT8	<b>FormatCount</b>	The number of Format handlers in a pipe
UINT8	<b>TaskMode</b>	The value indicating how Demuxer executes a pipe (See AMP_DEMUXER_TASK_MODE_e.)
UINT8	<b>Speed</b>	Demuxing speed (e.g., 1, 2, 4, 8, and 16)

Table 3-6. Definition of AMP\_DEMUXER\_PIPE\_CFG\_s for Demuxer API AmpDemuxer\_Create().

### 3.2.3 AmpDemuxer\_Delete

#### API Syntax:

**AmpDemuxer\_Delete** ( AMP\_DEMUXER\_PIPE\_HDLR\_s \* pipe )

#### Function Description:

- This function is used to delete a Demuxer pipe.

#### Parameters:

Type	Parameter	Description
AMP_DEMUXER_PIPE_HDLR_s *	<b>pipe</b>	The Demuxer pipe being deleted ( <a href="#">Section 3.2.1.1</a> )

Table 3-7. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_Delete()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-8. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_Delete()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 3.2.4 AmpDemuxer\_FeedFrame

### API Syntax:

**AmpDemuxer\_FeedFrame** ( AMP\_DMX\_FORMAT\_HDLR\_s \* format, UINT8 trackId, UINT32 targetTime, UINT8 frameType )

### Function Description:

- This function is used to feed a frame into a FIFO.

### Parameters:

Type	Parameter	Description
AMP_DMX_FORMAT_HDLR_s *	<b>format</b>	The Format handler ( <a href="#">Section 5.4.1.2</a> )
UINT8	<b>trackId</b>	The ID of a track that the new frame is fed into its FIFO
UINT32	<b>targetTime</b>	The reference time of the frame
UINT8	<b>frameType</b>	The type of the frame

Table 3-9. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_FeedFrame()**.

### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-10. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_FeedFrame()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.5 AmpDemuxer\_GetDefaultCfg

#### API Syntax:

**AmpDemuxer\_GetDefaultCfg** ( AMP\_DEMUXER\_PIPE\_CFG\_s \* config )

#### Function Description:

- This function is used to get the default configuration for initializing Demuxer pipes.

#### Parameters:

Type	Parameter	Description
AMP_DEMUXER_PIPE_CFG_s *	<b>config</b>	The returned configuration ( <a href="#">Section 3.2.2.1</a> )

Table 3-11. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_GetDefaultCfg()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-12. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_GetDefaultCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.6 AmpDemuxer\_GetDefaultImageInfoCfg

#### API Syntax:

**AmpDemuxer\_GetDefaultImageInfoCfg** ( AMP\_DMX\_IMAGE\_INFO\_CFG\_s \* config, AMP\_IMAGE\_INFO\_s \* image )

#### Function Description:

- This function is used to get the default configuration of an Image information object for demuxing.

#### Parameters:

Type	Parameter	Description
AMP_DMX_IMAGE_INFO_CFG_s *	<b>config</b>	The returned configuration ( <a href="#">Chapter 3.2.6.1</a> )
AMP_IMAGE_INFO_s *	<b>image</b>	The image information object being referred ( <a href="#">Section 5.2.1.1</a> )

Table 3-13. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_GetDefaultImageInfoCfg()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-14. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_GetDefaultImageInfoCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

#### 3.2.6.1 AmpDemuxer\_GetDefaultImageInfoCfg > AMP\_DMX\_IMAGE\_INFO\_CFG\_s

Type	Field	Description
AMP_FIFO_HDLR_s *	<b>Fifo</b>	The FIFO handler of image frames (Each Image Information object has an individual FIFO handler.) AMP_FIFO_HDLR_s is defined in Fifo.h, please refer to <a href="#">Section 3.2.6.2</a> )
UINT8 *	<b>BufferBase</b>	The start address of a FIFO buffer (Users push data into a FIFO; the FIFO will write the data to its buffer.)
UINT8 *	<b>BufferLimit</b>	The end address of a FIFO buffer (FIFO size = FIFO buffer limit - FIFO buffer base)

Table 3-15. Definition of **AMP\_DMX\_IMAGE\_INFO\_CFG\_s** for Demuxer API **AmpDemuxer\_GetDefaultImageInfoCfg()**.

### 3.2.6.2 AmpDemuxer\_GetDefaultImageInfoCfg > AMP\_FIFO\_HDLR\_s

Type	Field	Description
UINT32	<b>nFifold</b>	The unique of the FIFO
void*	<b>Ctx</b>	Private data of the FIFO

Table 3-16. Definition of **AMP\_FIFO\_HDLR\_s** for Demuxer API **AmpDemuxer\_GetDefaultImageInfoCfg()**.

Confidential  
For HAOTEK Only



### 3.2.7 AmpDemuxer\_GetDefaultMovieInfoCfg

#### API Syntax:

**AmpDemuxer\_GetDefaultMovieInfoCfg** ( AMP\_DMX\_MOVIE\_INFO\_CFG\_s \* config, AMP\_MOVIE\_INFO\_s \* movie )

#### Function Description:

- This function is used to get the default configuration of a Movie Information object for demuxing.

#### Parameters:

Type	Parameter	Description
AMP_DMX_MOVIE_INFO_CFG_s *	<b>config</b>	The returned configuration ( <a href="#">Section 3.2.7.1</a> )
AMP_MOVIE_INFO_s *	<b>movie</b>	The Movie Information object being referred ( <a href="#">Section 5.2.2.1</a> )

Table 3-17. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_GetDefaultMovieInfoCfg()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-18. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_GetDefaultMovieInfoCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.7.1 AmpDemuxer\_GetDefaultMovieInfoCfg > AMP\_DMX\_MOVIE\_INFO\_CFG\_s

Type	Field	Description
AMP_DMX_MEDIA_TRACK_CFG_s [AMP_FORMAT_MAX_TRACK_PER_MEDIA]	Track	Track configurations ( <a href="#">Section 3.2.7.2</a> )
UINT32	InitTime	The initial time (ms) of the media

Table 3-19. Definition of **AMP\_DMX\_MOVIE\_INFO\_CFG\_s** for Demuxer API **AmpDemuxer\_GetDefaultMovieInfoCfg()**.

### 3.2.7.2 AmpDemuxer\_GetDefaultMovieInfoCfg > AMP\_DMX\_MEDIA\_TRACK\_CFG\_s

Type	Field	Description
AMP_FIFO_HDLR_s *	Fifo	The FIFO handler of a track (Each track has an individual FIFO handler.) (Please refer to <a href="#">Section 3.2.6.2</a> )
UINT8 *	BufferBase	The start address of a FIFO buffer (Users push data into a FIFO; the FIFO will write the data to its buffer.)
UINT8 *	BufferLimit	The end address of a FIFO buffer (FIFO size = FIFO buffer limit - FIFO buffer base)
union { AMP_DMX_VIDEO_TRACK_CFG_s Video; AMP_DMX_AUDIO_TRACK_CFG_s Audio; AMP_DMX_TEXT_TRACK_CFG_s Text; }	Info	Video: Information of a video track ( <a href="#">Section 3.2.7.3</a> ) Audio: Information of an audio track ( <a href="#">Section 3.2.7.4</a> ) Text: Information of a text track ( <a href="#">Section 3.2.7.5</a> )

Table 3-20. Definition of **AMP\_DMX\_MEDIA\_TRACK\_CFG\_s** for Demuxer API **AmpDemuxer\_GetDefaultMovieInfoCfg()**.

### 3.2.7.3 AmpDemuxer\_GetDefaultMovieInfoCfg > AMP\_DMX\_VIDEO\_TRACK\_CFG\_s

Type	Field	Description
UINT8[4]	Resv	Reserved

Table 3-21. Definition of **AMP\_DMX\_VIDEO\_TRACK\_CFG\_s** for Demuxer API **AmpDemuxer\_GetDefaultMovieInfoCfg()**.

### 3.2.7.4 AmpDemuxer\_GetDefaultMovieInfoCfg > AMP\_DMX\_AUDIO\_TRACK\_CFG\_s

Type	Field	Description
UINT8[4]	Resv	Reserved

Table 3-22. Definition of **AMP\_DMX\_AUDIO\_TRACK\_CFG\_s** for Demuxer API **AmpDemuxer\_GetDefaultMovieInfoCfg()**.

### 3.2.7.5 AmpDemuxer\_GetDefaultMovieInfoCfg > AMP\_DMX\_TEXT\_TRACK\_CFG\_s

Type	Field	Description
UINT8[4]	Resv	Reserved

Table 3-23. Definition of **AMP\_DMX\_TEXT\_TRACK\_CFG\_s** for Demuxer API **AmpDemuxer\_GetDefaultMovieInfoCfg()**.

Confidential  
For HAOTEK Only

### 3.2.8 AmpDemuxer\_GetDefaultSoundInfoCfg

#### API Syntax:

**AmpDemuxer\_GetDefaultSoundInfoCfg** ( AMP\_DMX\_SOUND\_INFO\_CFG\_s \* config, AMP\_SOUND\_INFO\_s \* sound )

#### Function Description:

- This function is used to get the default configuration of a Sound information object for demuxing.

#### Parameters:

Type	Parameter	Description
AMP_DMX_SOUND_INFO_CFG_s *	<b>config</b>	The returned configuration ( <a href="#">Section 3.2.8.1</a> )
AMP_SOUND_INFO_s *	<b>sound</b>	The Sound information object being referred ( <a href="#">Section 5.2.3.1</a> )

Table 3-24. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_GetDefaultSoundInfoCfg()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-25. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_GetDefaultSoundInfoCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

#### 3.2.8.1 AmpDemuxer\_GetDefaultSoundInfoCfg > AMP\_DMX\_SOUND\_INFO\_CFG\_s

Type	Field	Description
AMP_DMX_MEDIA_TRACK_CFG_s [AMP_FORMAT_MAX_TRACK_PER_MEDIA]	<b>Track</b>	Track configuration ( <a href="#">Section 3.2.7.2</a> )
UINT32	<b>InitTime</b>	The initial time (ms) of the media.

Table 3-26. Definition of **AMP\_DMX\_SOUND\_INFO\_CFG\_s** for Demuxer API **AmpDemuxer\_GetDefaultSoundInfoCfg()**.

### 3.2.9 AmpDemuxer\_GetInitDefaultCfg

#### API Syntax:

**AmpDemuxer\_GetInitDefaultCfg** ( AMP\_DEMUXER\_INIT\_CFG\_s \* config )

#### Function Description:

- This function is used to get the default configuration for initializing the Demuxer module.

#### Parameters:

Type	Parameter	Description
AMP_DEMUXER_INIT_CFG_s *	<b>config</b>	The returned configuration ( <a href="#">Section 3.2.9.1</a> )

Table 3-27. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_GetInitDefaultCfg()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-28. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_GetInitDefaultCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.9.1 AmpDemuxer\_GetInitDefaultCfg > AMP\_DEMUXER\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the Demuxer module
UINT32	<b>BufferSize</b>	The size of the work buffer
AMP_TASK_INFO_s	<b>TaskInfo</b>	The information of a Demuxer task ( <a href="#">Section 3.2.9.2</a> )
UINT8	<b>MaxPipe</b>	The maximum number of pipes held in the Demuxer module
UINT8	<b>MaxTask</b>	The maximum number of tasks held in the Demuxer module

Table 3-29. Definition of **AMP\_DEMUXER\_INIT\_CFG\_s** for Demuxer API **AmpDemuxer\_GetInitDefaultCfg()**.

### 3.2.9.2 AmpDemuxer\_GetInitDefaultCfg > AMP\_TASK\_INFO\_s

Type	Field	Description
UINT32	<b>Priority</b>	Task priority
UINT32	<b>StackSize</b>	Stack size of tasks
UINT32	<b>CoreSelection</b>	Core selection bitmap. Bit[0] = 1 means core#0 is selected.

Table 3-30. Definition of **AMP\_TASK\_INFO\_s** for Demuxer API **AmpDemuxer\_GetInitDefaultCfg()**.

### 3.2.10 AmpDemuxer\_GetRequiredBufferSize

#### API Syntax:

**AmpDemuxer\_GetRequiredBufferSize** ( UINT8 maxPipe, UINT8 maxTask, UINT32 stackSize )

#### Function Description:

- This function is used to get the required buffer size for initializing the Demuxer module.

#### Parameters:

Type	Parameter	Description
UINT8	<b>maxPipe</b>	The maximum number of Demuxer pipes
UINT8	<b>maxTask</b>	The maximum number of Demuxer tasks
UINT32	<b>stackSize</b>	The stack size of each task (bytes)

Table 3-31. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_GetRequiredBufferSize()**.

#### Returns:

Return	Description
Size	The required buffer size

Table 3-32. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_GetRequiredBufferSize()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.11 AmpDemuxer\_Init

#### API Syntax:

**AmpDemuxer\_Init** ( AMP\_DEMUXER\_INIT\_CFG\_s \* config )

#### Function Description:

- This function is used to initialize the Demuxer module.

#### Parameters:

Type	Parameter	Description
AMP_DEMUXER_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the Demuxer module ( <a href="#">Section 3.2.9.1</a> )

Table 3-33. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_Init()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-34. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_Init()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



### 3.2.12 AmpDemuxer\_InitImageInfo

#### API Syntax:

**AmpDemuxer\_InitImageInfo** (AMP\_IMAGE\_INFO\_s \* image, AMP\_DMX\_IMAGE\_INFO\_CFG\_s \* config)

#### Function Description:

- This function is used to initialize an Image information object.

#### Parameters:

Type	Parameter	Description
AMP_IMAGE_INFO_s *	<b>image</b>	The Image information object being initialized ( <a href="#">Section 5.2.1.1</a> )
AMP_DMX_IMAGE_INFO_CFG_s *	<b>config</b>	The configuration used to initialize the Image information object ( <a href="#">Section 3.2.6.1</a> )

Table 3-35. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_InitImageInfo()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-36. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_InitImageInfo()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.13 AmpDemuxer\_InitMovieInfo

#### API Syntax:

```
AmpDemuxer_InitMovieInfo ( AMP_MOVIE_INFO_s * movie, AMP_DMX_MOVIE_INFO_CFG_s * config )
```

#### Function Description:

- This function is used to initialize a Movie information object.

#### Parameters:

Type	Parameter	Description
AMP_MOVIE_INFO_s *	<b>movie</b>	The Movie information object being initialized ( <a href="#">Section 5.2.2.1</a> )
AMP_DMX_MOVIE_INFO_CFG_s *	<b>config</b>	The configuration used to initialize the Movie information object ( <a href="#">Section 3.2.7.1</a> )

Table 3-37. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_InitMovieInfo()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-38. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_InitMovieInfo()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.14 AmpDemuxer\_InitSoundInfo

#### API Syntax:

**AmpDemuxer\_InitSoundInfo** ( AMP\_SOUND\_INFO\_s \* sound, AMP\_DMX\_SOUND\_INFO\_CFG\_s \* config )

#### Function Description:

- This function is used to initialize a Sound information object.

#### Parameters:

Type	Parameter	Description
AMP_SOUND_INFO_s *	<b>sound</b>	The Sound information object being initialized ( <a href="#">Section 5.2.3.1</a> )
AMP_DMX_SOUND_INFO_CFG_s *	<b>config</b>	The configuration used to initialize the Sound information object ( <a href="#">Section 3.2.8.1</a> )

Table 3-39. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_InitSoundInfo()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-40. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_InitSoundInfo()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.15 AmpDemuxer\_OnDataRequest

#### API Syntax:

**AmpDemuxer\_OnDataRequest** ( AMP\_FIFO\_HDLR\_s \* fifo )

#### Function Description:

- This function is used to request Demuxer to feed new frames into a FIFO.

#### Parameters:

Type	Parameter	Description
AMP_FIFO_HDLR_s *	<b>fifo</b>	The FIFO to which new frames are fed. (Please refer to <a href="#">Section 3.2.6.2</a> )

Table 3-41. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_OnDataRequest()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-42. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_OnDataRequest()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.16 AmpDemuxer\_Remove

#### API Syntax:

**AmpDemuxer\_Remove** (AMP\_DEMUXER\_PIPE\_HDLR\_s \* pipe)

#### Function Description:

- This function is used to remove a Demuxer pipe.

#### Parameters:

Type	Parameter	Description
AMP_DEMUXER_PIPE_HDLR_s *	<b>pipe</b>	The Demuxer pipe being removed ( <a href="#">Section 3.2.1.1</a> )

Table 3-43. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_Remove()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-44. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_Remove()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.17 AmpDemuxer\_Seek

#### API Syntax:

**AmpDemuxer\_Seek** ( AMP\_DEMUXER\_PIPE\_HDLR\_s \* pipe, UINT32 targetTime, UINT8 direction, UINT32 speed )

#### Function Description:

- This function is used to seek and set the start time of demuxing.

#### Parameters:

Type	Parameter	Description
AMP_DEMUXER_PIPE_HDLR_s *	<b>pipe</b>	The Demuxer pipe ( <a href="#">Section 3.2.1.1</a> )
UINT32	<b>targetTime</b>	The reference time being sought
UINT8	<b>direction</b>	Seek direction
UINT32	<b>speed</b>	The demuxing speed after seeking.

Table 3-45. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_Seek()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-46. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_Seek()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.18 AmpDemuxer\_SetProcParam

#### API Syntax:

**AmpDemuxer\_SetProcParam** ( AMP\_DEMUXER\_PIPE\_HDLR\_s \* pipe, UINT32 procParam )

#### Function Description:

- This function is used to set the process parameter of a Demuxer pipe.

#### Parameters:

Type	Parameter	Description
AMP_DEMUXER_PIPE_HDLR_s *	<b>pipe</b>	The pipe applying the parameters ( <a href="#">Section 3.2.1.1</a> )
UINT32	<b>procParam</b>	The process parameter

Table 3-47. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_SetProcParam()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-48. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_SetProcParam()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.19 AmpDemuxer\_Start

#### API Syntax:

**AmpDemuxer\_Start** ( AMP\_DEMUXER\_PIPE\_HDLR\_s \* pipe )

#### Function Description:

- This function is used to start a Demuxer pipe.

#### Parameters:

Type	Parameter	Description
AMP_DEMUXER_PIPE_HDLR_s *	<b>pipe</b>	The Demuxer pipe being started ( <a href="#">Section 3.2.1.1</a> )

Table 3-49. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_Start()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-50. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_Start()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



### 3.2.20 AmpDemuxer\_Stop

#### API Syntax:

**AmpDemuxer\_Stop** ( AMP\_DEMUXER\_PIPE\_HDLR\_s \* pipe )

#### Function Description:

- This function is used to stop a Demuxer pipe.

#### Parameters:

Type	Parameter	Description
AMP_DEMUXER_PIPE_HDLR_s *	<b>pipe</b>	The Demuxer pipe being stopped ( <a href="#">Section 3.2.1.1</a> )

Table 3-51. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_Stop()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-52. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_Stop()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 3.2.21 AmpDemuxer\_WaitComplete

#### API Syntax:

**AmpDemuxer\_WaitComplete** ( AMP\_DEMUXER\_PIPE\_HDLR\_s \* pipe, UINT32 timeOut )

#### Function Description:

- This function is used to poll the status of a pipe to check whether its life cycle is complete or not.

#### Parameters:

Type	Parameter	Description
AMP_DEMUXER_PIPE_HDLR_s *	<b>pipe</b>	The pipe being polled ( <a href="#">Section 3.2.1.1</a> )
UINT32	<b>timeOut</b>	The polling interval (ms)

Table 3-53. Parameters for SDK6 API Middleware Demuxer API **AmpDemuxer\_WaitComplete()**.

#### Returns:

Return	Description
0	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 3-54. Returns for SDK6 Middleware Demuxer API **AmpDemuxer\_WaitComplete()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

# 4 Editor

## 4.1 Editor: Overview

Editor implements the APIs for users to edit files. It includes the following functions:

1. Initialize the Editor module
2. Crop2New function
3. Divide function
4. Merge function
5. Partial Delete function
6. Recover function

## 4.2 Editor: List of APIs

- [AmpEditor\\_Abort](#)
- [AmpEditor\\_Crop2New](#)
- [AmpEditor\\_Divide](#)
- [AmpEditor\\_GetInitDefaultCfg](#)
- [AmpEditor\\_Init](#)
- [AmpEditor\\_Merge](#)
- [AmpEditor\\_PartialDelete](#)
- [AmpEditor\\_Recover](#)

## 4.2.1 AmpEditor\_Abort

### API Syntax:

**AmpEditor\_Abort** (AMP\_EDT\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to abort the Editor operation.

### Parameters:

Type	Parameter	Description
AMP_EDT_FORMAT_HDLR_s *	hdlr	The format handler used by Editor. Please refer to <a href="#">Section 4.2.2.1</a> for more details.

Table 4-1. Parameters for Editor SDK6 API **AmpEditor\_Abort()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 4-2. Returns for Editor SDK6 API **AmpEditor\_Abort()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 4.2.2 AmpEditor\_Crop2New

### API Syntax:

**AmpEditor\_Crop2New** (AMP\_EDT\_FORMAT\_HDLR\_s \* hdlr, BOOL newTask, UINT32 timeStart, UINT32 timeEnd, BOOL trim, char \* fileNameIn, char \* fileNameOut)

### Function Description:

- This function is used to crop a segment of an input file to create a new one.

### Parameters:

Type	Parameter	Description
AMP_EDT_FORMAT_HDLR_s *	<b>hdlr</b>	The format handler used by Editor. Please refer to <a href="#">Section 4.2.2.1</a> for more details.
BOOL	<b>newTask</b>	The value is used to indicate that the editing operation is working on a new task or the main task.
UINT32	<b>timeStart</b>	The start time of the cropped segment (ms)
UINT32	<b>timeEnd</b>	The end time of the cropped segment (ms)
BOOL	<b>trim</b>	The flag used to enable Editor to trim the output file
char *	<b>fileNameIn</b>	The name of the input file
char *	<b>fileNameOut</b>	The name of the output file

Table 4-3. Parameters for Editor SDK6 API **AmpEditor\_Crop2New()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 4-4. Returns for Editor SDK6 API **AmpEditor\_Crop2New()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



Type	Field	Description
int(*) (AMP_EDT_FORMAT_HDLR_s*, AMP_STREAM_HDLR_s*, AMP_MEDIA_INFO_s*, UINT32)	<b>Merge</b>	The interface to prepare the materials required by merging two input files into two parts according to a specified time (Handler, Divide time, Trim flag, Round mode, Original file stream, Original media information, New media information, New media information, alignment flag)
int(*) (AMP_EDT_FORMAT_HDLR_s*, AMP_STREAM_HDLR_s*, AMP_MEDIA_INFO_s*)	<b>Recover</b>	The interface to prepare the materials required by recovering a bad or abnormal closed recording file (Handler, File stream, Trim flag, File name)
int(*) (AMP_EDT_FORMAT_HDLR_s*, AMP_STREAM_HDLR_s*, AMP_MEDIA_INFO_s*)	<b>Finalize</b>	The interface to finalize the specified media file (Handler, File stream, Media information)
int(*) (AMP_EDT_FORMAT_HDLR_s*, UINT32, void*, AMP_STREAM_HDLR_s*, AMP_MEDIA_INFO_s*)	<b>UpdateAtom</b>	The interface to update an atom (Handler, Atom Id, Parameter, File stream, Media information)
int(*) (AMP_EDT_FORMAT_HDLR_s*, UINT32, void*)	<b>Func</b>	The interface to execute special commands (Handler, Command, Parameter)

Table 4-6. Definition of **AMP\_EDT\_FORMAT\_s** for Editor SDK6 API **AmpEditor\_Crop2New()**.

### 4.2.3 AmpEditor\_Divide

#### API Syntax:

**AmpEditor\_Divide** (AMP\_EDT\_FORMAT\_HDLR\_s \* hdlr, BOOL newTask, UINT32 targetTime, BOOL trim, UINT8 roundMode, char \* fileNameIn, char \* fileNameOut)

#### Function Description:

- This function is used to divide an input file into two parts according to a specified time. (The first part is retained in the input file, and the second one is stored in an output file)

#### Parameters:

Type	Parameter	Description
AMP_EDT_FORMAT_HDLR_s *	<b>hdlr</b>	The Format handler used by the Editor. Please refer to <a href="#">Section 4.2.2.1</a> for more details.
BOOL	<b>newTask</b>	The value is used to indicate that the editing operation is working on a new task or the main task.
UINT32	<b>targetTime</b>	The reference time to dividing the input file (ms)
BOOL	<b>trim</b>	The flag used to enable Editor to trim the output file
UINT8	<b>roundMode</b>	The value used to indicate a method to determine a real dividing time (Please refer to AMP_EDITOR_ROUND_MODE_e)
char *	<b>fileNameIn</b>	The name of the input file
char *	<b>fileNameOut</b>	The name of the output file

Table 4-7. Parameters for Editor SDK6 API **AmpEditor\_Divide()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 4-8. Returns for Editor SDK6 API **AmpEditor\_Divide()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 4.2.4 AmpEditor\_GetInitDefaultCfg

### API Syntax:

**AmpEditor\_GetInitDefaultCfg** (AMP\_EDITOR\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the Editor module.

### Parameters:

Type	Parameter	Description
AMP_EDITOR_INIT_CFG_s *	<b>config</b>	The returned configuration. Please refer to <a href="#">Section 4.2.4.1</a> for more details.

Table 4-9. Parameters for Editor SDK6 API **AmpEditor\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 4-10. Returns for Editor SDK6 API **AmpEditor\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 4.2.4.1 AmpEditor\_GetInitDefaultCfg > AMP\_EDITOR\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of Editor
UINT32	<b>BufferSize</b>	The size of the work buffer
AMP_TASK_INFO_s	<b>TaskInfo</b>	The task information ( <a href="#">Section 2.2.13.2</a> )
UINT8	<b>MaxCmd</b>	The maximum number of commands queued in Editor
UINT8	<b>MaxSubTask</b>	The maximum number of sub tasks held in Editor

Table 4-11. Definition of **AMP\_EDITOR\_INIT\_CFG\_s** for Editor API **AmpEditor\_GetInitDefaultCfg()**.

## 4.2.5 AmpEditor\_Init

### API Syntax:

**AmpEditor\_Init** (AMP\_EDITOR\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initialize the Editor module.

### Parameters:

Type	Parameter	Description
AMP_EDITOR_INIT_CFG_s *	<b>config</b>	This is the configuration for initializing the Editor module. Please refer to <a href="#">Section 4.2.4.1</a> for more details.

Table 4-12. Parameters for Editor SDK6 API **AmpEditor\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 4-13. Returns for Editor SDK6 API **AmpEditor\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 4.2.6 AmpEditor\_Merge

### API Syntax:

**AmpEditor\_Merge** (AMP\_EDT\_FORMAT\_HDLR\_s \* hdlr, BOOL newTask, BOOL trim, BOOL directed, char \* fileNameIn1, char \* fileNameIn2)

### Function Description:

- This function is used to merge two input files into a single one. The second input file will be appended to the first one.

### Parameters:

Type	Parameter	Description
AMP_EDT_FORMAT_HDLR_s *	<b>hdlr</b>	The format handler used by the Editor. Please refer to <a href="#">Section 4.2.2.1</a> for more details.
BOOL	<b>newTask</b>	The value is used to indicate that the file operation is worked on a new task or the main task.
BOOL	<b>trim</b>	The flag is used to enable Editor to trim the output file.
BOOL	<b>directed</b>	The flag to enable Editor to directly append the second input file to the first one without any silence frames.
char*	<b>fileNameIn1</b>	The name of the first input file
char*	<b>fileNameIn2</b>	The name of the second input file

Table 4-14. Parameters for Editor SDK6 API **AmpEditor\_Merge()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e . Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 4-15. Returns for Editor SDK6 API **AmpEditor\_Merge()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 4.2.7 AmpEditor\_PartialDelete

### API Syntax:

**AmpEditor\_PartialDelete** (AMP\_EDT\_FORMAT\_HDLR\_s \* hdlr, BOOL newTask, UINT32 timeStart, UINT32 timeEnd, BOOL trim, char \* fileNameIn, char \* fileNameOut)

### Function Description:

- This function is used to delete a segment of an input file and change the file name to fileNameOut.

### Parameters:

Type	Parameter	Description
AMP_EDT_FORMAT_HDLR_s *	<b>hdlr</b>	The Format handler used by Editor. Please refer to <a href="#">Section 4.2.2.1</a> for more details.
BOOL	<b>newTask</b>	The value used to indicate that the file operation is working on a new task or the main task
UINT32	<b>timeStart</b>	The start time of the removed segment (ms)
UINT32	<b>timeEnd</b>	The end time of the removed segment (ms)
BOOL	<b>trim</b>	The flag used to enable Editor to trim the output file
char *	<b>fileNameIn</b>	The name of the input file
char *	<b>fileNameOut</b>	The name of the output file

Table 4-16. Parameters for Editor SDK6 API **AmpEditor\_PartialDelete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 4-17. Returns for Editor SDK6 API **AmpEditor\_PartialDelete()**.

### Example

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 4.2.8 AmpEditor\_Recover

### API Syntax:

**AmpEditor\_Recover** (AMP\_EDT\_FORMAT\_HDLR\_s \* hdlr, BOOL newTask, char \* fileName)

### Function Description:

- This function is used to recover a bad or abnormally closed recording file.

### Parameters:

Type	Parameter	Description
AMP_EDT_FORMAT_HDLR_s *	<b>hdlr</b>	The Format handler used by Editor. Please refer to <a href="#">Section 4.2.2.1</a> for more details.
BOOL	<b>newTask</b>	The value is used to indicate that the file operation is worked on a new task or the main task
char *	<b>fileName</b>	The name of the input file being recovered

Table 4-18. Parameters for Editor SDK6 API **AmpEditor\_Recover()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 4-19. Returns for Editor SDK6 API **AmpEditor\_Recover()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

# 5 Format

## 5.1 Format: Overview

This chapter provides Turn-Keys for Format. Format is the Media container implementation. Format includes the following sections:

- Dummy
- EXIF
- EXT
- Matroska (.MKV)
- MOV
- MP4

## 5.2 Format: List of APIs

Format:

- [AmpFormat\\_CopyImageInfo](#)
- [AmpFormat\\_CopyMovieInfo](#)
- [AmpFormat\\_CopySoundInfo](#)
- [AmpFormat\\_FlushMediaInfo](#)
- [AmpFormat\\_GetImageInfo](#)
- [AmpFormat\\_GetInitDefaultCfg](#)
- [AmpFormat\\_GetMovieInfo](#)
- [AmpFormat\\_GetRequiredBufferSize](#)
- [AmpFormat\\_GetSoundInfo](#)
- [AmpFormat\\_Init](#)
- [AmpFormat\\_NewImageInfo](#)
- [AmpFormat\\_NewMovieInfo](#)
- [AmpFormat\\_NewSoundInfo](#)
- [AmpFormat\\_RelImageInfo](#)
- [AmpFormat\\_RelMovieInfo](#)
- [AmpFormat\\_RelSoundInfo](#)

Dummy:

- [AmpDummyMux\\_Create](#)
- [AmpDummyMux\\_Delete](#)
- [AmpDummyMux\\_GetDefaultCfg](#)
- [AmpDummyMux\\_GetInitDefaultCfg](#)
- [AmpDummyMux\\_GetRequiredBufferSize](#)
- [AmpDummyMux\\_Init](#)

Exif:

- AmpExifDmx\_Create
- AmpExifDmx\_Delete
- AmpExifDmx\_GetDefaultCfg
- AmpExifDmx\_GetInitDefaultCfg
- AmpExifDmx\_GetRequiredBufferSize
- AmpExifDmx\_Init
- AmpExifDmx\_Parse
- AmpExifMux\_Create
- AmpExifMux\_Delete
- AmpExifMux\_GetDefaultCfg
- AmpExifMux\_GetInitDefaultCfg
- AmpExifMux\_GetRequiredBufferSize
- AmpExifMux\_Init

Ext:

- AmpExtDmx\_Create
- AmpExtDmx\_Delete
- AmpExtDmx\_GetDefaultCfg
- AmpExtDmx\_GetInitDefaultCfg
- AmpExtDmx\_GetRequiredBufferSize
- AmpExtDmx\_Init
- AmpExtDmx\_Parse
- AmpExtMux\_Create
- AmpExtMux\_Delete
- AmpExtMux\_GetDefaultCfg
- AmpExtMux\_GetInitDefaultCfg
- AmpExtMux\_GetRequiredBufferSize
- AmpExtMux\_Init

Mkv:

- AmpMkvDmx\_Create
- AmpMkvDmx\_Delete
- AmpMkvDmx\_GetDefaultCfg
- AmpMkvMux\_GetInitDefaultCfg
- AmpMkvDmx\_GetRequiredBufferSize
- AmpMkvDmx\_Init
- AmpMkvDmx\_Parse
- AmpMkvEdt\_Create
- AmpMkvEdt\_Delete
- AmpMovEdt\_GetDefaultCfg
- AmpMkvEdt\_GetInitDefaultCfg
- AmpMkvEdt\_GetRequiredBufferSize
- AmpMkvEdt\_Init
- AmpMkvMux\_Create
- AmpMkvMux\_Delete
- AmpMkvMux\_GetDefaultCfg
- AmpMkvMux\_GetInitDefaultCfg
- AmpMkvMux\_GetRequiredBufferSize

- AmpMkvMux\_Init

Mov:

- AmpMovDmx\_Create
- AmpMovDmx\_Delete
- AmpMovDmx\_GetDefaultCfg
- AmpMovDmx\_GetInitDefaultCfg
- AmpMovDmx\_GetRequiredBufferSize
- AmpMovDmx\_Init
- AmpMovDmx\_Parse
- AmpMovEdt\_Create
- AmpMovEdt\_Delete
- AmpMovMux\_GetDefaultCfg
- AmpMovEdt\_GetInitDefaultCfg
- AmpMovEdt\_GetRequiredBufferSize
- AmpMovEdt\_Init
- AmpMovMux\_Create
- AmpMovMux\_Delete
- AmpMovMux\_GetDefaultCfg
- AmpMovMux\_GetInitDefaultCfg
- AmpMovEdt\_GetRequiredBufferSize
- AmpMovDmx\_Init

MP4:

- AmpMp4Dmx\_GetDefaultCfg
- AmpMp4Dmx\_GetInitDefaultCfg
- AmpMp4Dmx\_GetRequiredBufferSize
- AmpMp4Dmx\_Init
- AmpMp4Dmx\_Parse
- AmpMp4Edt\_Create
- AmpMp4Edt\_Delete
- AmpMp4Edt\_GetDefaultCfg
- AmpMp4Edt\_GetInitDefaultCfg
- AmpMp4Edt\_GetRequiredBufferSize
- AmpMp4Edt\_Init
- AmpMp4Mux\_Create
- AmpMp4Mux\_Delete
- AmpMp4Mux\_GetDefaultCfg
- AmpMp4Mux\_GetInitDefaultCfg
- AmpMp4Mux\_GetRequiredBufferSize
- AmpMp4Mux\_Init



## 5.2.1 AmpFormat\_CopyImageInfo

### API Syntax:

**AmpFormat\_CopyImageInfo** (AMP\_IMAGE\_INFO\_s \* dstInfo, AMP\_IMAGE\_INFO\_s \* srcInfo)

### Function Description:

- This function is used to copy an image information object to a destination object.

### Parameters:

Type	Parameter	Description
AMP_IMAGE_INFO_s *	<b>dstInfo</b>	The destination image information object. ( <b>AMP_IMAGE_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to below <a href="#">Section 5.2.1.1</a> below for definition.
AMP_IMAGE_INFO_s *	<b>srcInfo</b>	The source image information object. Please refer to <a href="#">Section 5.2.1.1</a> below for more details.

Table 5-1. Parameters for Format SDK6 API **AmpFormat\_CopyImageInfo()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-2. Returns for Format SDK6 API **AmpFormat\_CopyImageInfo()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.2.1.1 AmpFormat\_CopyImageInfo > AMP\_IMAGE\_INFO\_s

Type	Field	Description
UINT8	<b>MediaType</b>	The media type (Please refer to AMP_MEDIA_INFO_TYPE_e)
BOOL8	<b>Valid</b>	The value indicating if the media context is valid
UINT8	<b>SubFormat</b>	The sub_format of the media (MSMP4, FUJIMOV)
UINT32	<b>MagicPattern</b>	The Magic pattern 0x12345678 (used for recovery)
UINT64	<b>Size</b>	The file size
char [MAX_FILENAME_LENGTH]	<b>Name</b>	The media name
char [AMP_FORMAT_MAX_DATE_SIZE]	<b>Date</b>	The creation date of the media
char [AMP_FORMAT_MAX_TIME_SIZE]	<b>Time</b>	The creation time of the media
AMP_FIFO_HDLR_s *	<b>Fifo</b>	The FIFO handler of an image (Each image has an individual FIFO handler.) (Please refer to <a href="#">Section 3.2.6.2</a> )
UINT8 *	<b>BufferBase</b>	The start address of the FIFO buffer (Users push data into the FIFO. The FIFO will write data to the address of the buffer.
UINT8 *	<b>BufferLimit</b>	The end address of the FIFO buffer (FIFO size = FIFO buffer limit - FIFO buffer base)
AMP_IMAGE_FRAME_INFO_s [AMP_FORMAT_MAX_FRAME_PER_IMAGE]	<b>Frame</b>	The image frames (Please refer to <a href="#">Section 5.2.1.2</a> for more details)
union { AMP_EXT_PRIV_INFO_s Ext; AMP_EXIF_PRIV_INFO_s Exif; }	<b>PrivInfo</b>	Ext: The private data for the information of Ext Format ( <a href="#">Section 5.2.1.4</a> ) Exif: The private data for the information of Exif Format ( <a href="#">Section 5.2.1.5</a> )
UINT8	<b>UsedFrame</b>	The number of frames stored in an image file (the used entries of Frame [AMP_FORMAT_MAX_FRAME_PER_IMAGE]).
UINT8	<b>TotalFrame</b>	The total number of frames referred to by an image (Muxer only).
UINT8	<b>Endian</b>	The endian type of an image (e.g., big endian or little endian)

Table 5-3. Definition of AMP\_IMAGE\_INFO\_s for Format SDK6 API AmpFormat\_CopyImageInfo().

### 5.2.1.2 AmpFormat\_CopyImageInfo > AMP\_IMAGE\_FRAME\_INFO\_s

Type	Field	Description
UINT32	<b>SeqNum</b>	The sequence number of an image frame
EXIF_INFO_s	<b>ExifInfo</b>	Exif information (See EXIF_INFO_s.)
GPS_INFO_s	<b>GpsInfo</b>	GPS information (Please refer to <a href="#">Section 5.2.1.3</a> )
UINT32	<b>Type</b>	Image type
UINT32	<b>Pos</b>	Image position
UINT32	<b>Size</b>	Image size
UINT16	<b>Width</b>	Image width
UINT16	<b>Height</b>	Image height

Table 5-4. Definition of AMP\_IMAGE\_FRAME\_INFO\_s for Format SDK6 API AmpFormat\_CopyImageInfo().

### 5.2.1.3 AmpFormat\_CopyImageInfo > GPS\_INFO\_s

Type	Field	Description
UINT32	<b>VersionId</b>	GPSTimeStamp
UINT8[2]	<b>LatitudeRef</b>	GPSTimeStamp
UINT64[3]	<b>Latitude</b>	GPSTimeStamp
UINT8[2]	<b>LongitudeRef</b>	GPSTimeStamp
UINT64[3]	<b>Longitude</b>	GPSTimeStamp
UINT8	<b>AltitudeRef</b>	GPSTimeStamp
UINT64	<b>Altitude</b>	GPSTimeStamp
UINT64[3]	<b>Timestamp</b>	GPSTimeStamp
UINT32	<b>SatelliteOffset</b>	GPSSatellites tag offset
UINT32	<b>SatelliteCount</b>	GPSSatellites tag size
UINT8[2]	<b>Status</b>	GPSStatus
UINT8[2]	<b>MeasureMode</b>	GPSTimeStamp
UINT64	<b>Dop</b>	GPSTimeStamp
UINT8[2]	<b>SpeedRef</b>	GPSSpeedRef
UINT64	<b>Speed</b>	GPSSpeed
UINT8[2]	<b>TrackRef</b>	GPSTimeStamp
UINT64	<b>Track</b>	GPSTimeStamp
UINT8[2]	<b>ImgdirectionRef</b>	GPSTimeStamp
UINT64	<b>Imgdirection</b>	GPSTimeStamp
UINT32	<b>MapdatumOffset</b>	GPSTimeStamp
UINT32	<b>MapdatumCount</b>	GPSTimeStamp
UINT8[2]	<b>DestlatitudeRef</b>	GPSTimeStamp
UINT64[3]	<b>Destlatitude</b>	GPSTimeStamp
UINT8[2]	<b>DestlongitudeRef</b>	GPSTimeStamp
UINT64[3]	<b>Destlongitude</b>	GPSTimeStamp
UINT8[2]	<b>DestbearingRef</b>	GPSTimeStamp
UINT64	<b>Destbearing</b>	GPSTimeStamp
UINT8[2]	<b>DestdistanceRef</b>	GPSTimeStamp
UINT64	<b>Destdistance</b>	GPSTimeStamp

Type	Field	Description
UINT32	<b>ProcessingmethodOffset</b>	GPSProcessingMethod tag offset
UINT32	<b>ProcessingmethodCount</b>	GPSProcessingMethod tag size
UINT32	<b>AreainformationOffset</b>	GPSAreaInformation tag offset
UINT32	<b>AreainformationCount</b>	GPSAreaInformation tag size
UINT8[11]	<b>Datestamp</b>	GPSTimestamp
UINT16	<b>Differential</b>	GPSDifferential
UINT64	<b>HPositioningError</b>	GPSHPositioningError

Table 5-5. Definition of **GPS\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyImageInfo()**.

#### 5.2.1.4 AmpFormat\_CopyImageInfo > AMP\_EXT\_PRIV\_INFO\_s

Type	Field	Description
UINT8[4]	<b>Resv</b>	Reserve

Table 5-6. Definition of **AMP\_EXT\_PRIV\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyImageInfo()**.

#### 5.2.1.5 AmpFormat\_CopyImageInfo > AMP\_EXIF\_PRIV\_INFO\_s

Type	Field	Description
UINT32	<b>MakerNoteOff</b>	The start offset of makernote in the file
UINT32	<b>MakerNoteSize</b>	The size of makernote
UINT32	<b>App1Off</b>	The start offset of maker APP1 in the file
UINT32	<b>App2Off</b>	The start offset of maker APP2 in the file
UINT32	<b>TiffBase</b>	The start offset of Tiff header in the file
UINT32	<b>Ifd0Off</b>	The start offset of IFD0 structure in the file
UINT32	<b>Ifd1Off</b>	The start offset of IFD1 structure in the file
UINT32	<b>ExifIfdOff</b>	The start offset of ExifIfd structure in the file
UINT32	<b>GpsIfdOff</b>	The start offset of GpsIfd structure in the file
UINT32	<b>SecPicType</b>	The format type of the second picture (screenail)
BOOL8	<b>Thumbnail</b>	The flag of the thumbnail in the image
BOOL8	<b>Screenail</b>	The flag of the screenail in the image
BOOL8	<b>Fullview</b>	The flag of the fullview in the image

Table 5-7. Definition of **AMP\_EXIF\_PRIV\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyImageInfo()**.

## 5.2.2 AmpFormat\_CopyMovieInfo

### API Syntax:

**AmpFormat\_CopyMovieInfo** (AMP\_MOVIE\_INFO\_s \* dstInfo, AMP\_MOVIE\_INFO\_s \* srcInfo)

### Function Description:

- This function is used to copy a movie information object.

### Parameters:

Type	Parameter	Description
AMP_MOVIE_INFO_s *	<b>dstInfo</b>	The destination movie information object. ( <b>AMP_MOVIE_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.2.2.1</a> below for more details.
AMP_MOVIE_INFO_s *	<b>srcInfo</b>	The source movie information object. ( <b>AMP_MOVIE_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.2.2.1</a> below for more details.

Table 5-8. Parameters for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-9. Returns for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.2.2.1 AmpFormat\_CopyMovieInfo > AMP\_MOVIE\_INFO\_s

Type	Field	Description
UINT8	<b>MediaType</b>	The media type (Please refer to AMP_MEDIA_INFO_TYPE)
BOOL8	<b>Valid</b>	The value indicating if the media context is valid
UINT8	<b>SubFormat</b>	The sub format of the media (MSMP4, FUJIMOV)
UINT32	<b>MagicPattern</b>	The Magic pattern 0x12345678 (used for recovery)
UINT64	<b>Size</b>	The file size
char [MAX_FILENAME_LENGTH]	<b>Name</b>	The media name
char [AMP_FORMAT_MAX_DATE_SIZE]	<b>Date</b>	The creation date of the media
char [AMP_FORMAT_MAX_TIME_SIZE]	<b>Time</b>	The creation time of the media
AMP_MEDIA_TRACK_INFO_s [AMP_FORMAT_MAX_TRACK_PER_MEDIA]	<b>Track</b>	The tracks (Please refer to <a href="#">Section 5.2.2.2</a> )
union { AMP_MUX_PROC_INFO_s Mux; AMP_DMUX_PROC_INFO_s Dmx; }	<b>ProcInfo</b>	Mux: The muxing information ( <a href="#">Section 5.2.2.6</a> )  Dmx: The demuxing information ( <a href="#">Section 5.2.2.7</a> )
union { AMP_EXT_PRIV_INFO_s Ext; AMP_ISO_PRIV_INFO_s Iso; AMP_MATROSKA_PRIV_INFO_s Matroska; }	<b>PrivInfo</b>	Ext: The private data for the information of Ext format. (Section 5.2.1.4)  Iso: The private data for the information of Iso format. (Section 5.2.2.9)  Matroska: The private data for the information of Matroska format. ( <a href="#">Section 5.2.2.14</a> )
AMP_FIFO_HDLR_s *	<b>ThmFifo</b>	The FIFO handler for getting image thumbnails. (Please refer to <a href="#">Section 3.2.6.2</a> )
UINT8 *	<b>ThmBase</b>	The start address of the buffer for image thumbnails
UINT8 *	<b>ThmLimit</b>	The end address of the buffer for image thumbnails
AMP_THUMB_FRAME_INFO_s [AMP_FORMAT_MAX_FRAME_PER_IMAGE]	<b>Thumb</b>	The thumbnail frames (See <a href="#">Section 5.2.2.19</a> )

Type	Field	Description
UINT8	<b>TrackCount</b>	The number of tracks in a movie
UINT8	<b>ThumbCount</b>	The number of image thumbnail

Table 5-10. Definition of **AMP\_MOVIE\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

Confidential  
For HAOTEK Only

### 5.2.2.2 AmpFormat\_CopyMovieInfo > AMP\_MEDIA\_TRACK\_INFO\_s

Type	Field	Description
UINT32	<b>MediaId</b>	The media type of the track (The ID is a media ID. See AMP_FORMAT_MID_e)
UINT32	<b>TimeScale</b>	The ticks per second
UINT32	<b>OrigTimeScale</b>	The original TimeScale
UINT32	<b>TimePerFrame</b>	The ticks per frame
UINT32	<b>FrameNo</b>	Frame number (It is a logical number. In muxing or demuxing, the value is the frame number handled currently. If Demuxer is forward demuxing, the number will progressively increase. If Demuxer is backward demuxing, the number will progressively decrease)
UINT32	<b>FrameCount</b>	The count of frames in the track
UINT64	<b>InitDTS</b>	The initial value of the DTS (If a file is a split file, this specifies the start decode offset which the value generated by the previous clip)
UINT64	<b>DTS</b>	Decode time stamp (The value is based on the time scale of the track. In Muxer, the value is the DTS of next frame; in Demuxer, the value is the duration of the track)
UINT64	<b>NextDTS</b>	Next Decode time stamp (In Muxer, it predicates the DTS of the next frame. In Demuxer, the value is the same as DTS.)
AMP_FIFO_HDLR_s *	<b>Fifo</b>	The FIFO handler of the track (Each track has an individual FIFO handler) (Please refer to <a href="#">Section 3.2.6.2</a> )
UINT8 *	<b>BufferBase</b>	The start address of the FIFO buffer (Users push data to the FIFO, the FIFO will write the data to its buffer according to the address)
UINT8 *	<b>BufferLimit</b>	The end address of the FIFO buffer (FIFO size = FIFO buffer limit - FIFO buffer base)
union { AMP_VIDEO_TRACK_INFO_s Video; AMP_AUDIO_TRACK_INFO_s Audio; AMP_TEXT_TRACK_INFO_s Text; }	<b>Info</b>	Video: The information of the video track ( <a href="#">Section 5.2.2.3</a> )  Audio: The information of the video track ( <a href="#">Section 5.2.2.4</a> )  Text: The information of the video track ( <a href="#">Section 5.2.2.5</a> )
UINT8	<b>TrackId</b>	Track ID (Every track ID is unique)
UINT8	<b>TrackType</b>	Track type (See AMP_MEDIA_TRACK_TYPE_e)

Table 5-11. Definition of AMP\_MEDIA\_TRACK\_INFO\_s for Format SDK6 API AmpFormat\_CopyMovieInfo().



### 5.2.2.3 AmpFormat\_CopyMovieInfo > AMP\_VIDEO\_TRACK\_INFO\_s

Type	Field	Description
UINT32	<b>CodecTimeScale</b>	The time scale of a codec (The item is only used in AmpFormat_ConvertPTS.)
UINT32	<b>GOPSize</b>	Number of pictures between IDR pictures
UINT32	<b>FrameCountAfterResume</b>	The number of frames after resuming a video
UINT64	<b>RefDTS</b>	The DTS of the frame with PTS 0 (If a GOP has B frames, the DTS of the IDR frame is 0 references.)
UINT64	<b>InitPTS</b>	The DSP PTS of the first frame (The item is only used in AmpFormat_ConvertPTS.)
UINT64	<b>PTS</b>	Video PTS
UINT64	<b>DiffPTS</b>	For the case that the frame with the minimum PTS is B frame. (the item is only used in AmpFormat_ConvertPTS.)
UINT16	<b>PixelArX</b>	The aspect ratio X of the pixel (If the value is not 1, it means the pixel is not square.)
UINT16	<b>PixelArY</b>	The aspect ratio Y of the pixel (If the value is not 1, it means the pixel is not square.)
UINT16	<b>Width</b>	Video width
UINT16	<b>Height</b>	Video height
UINT16	<b>M</b>	The number of pictures between reference pictures (IDR, I, P)
UINT16	<b>N</b>	The number of pictures between I pictures
BOOL8	<b>IsDefault</b>	The flag indicating the track is the default video track
UINT8	<b>Mode</b>	The value indicating the picture mode of the video (It has progressive and interlaced modes. Interlaced mode has Field Per Sample and Frame Per Sample. See AMP_VIDEO_MODE_s.)
BOOL8	<b>ClosedGOP</b>	The flag indicating that the structure of a video track is closed GOP (The structure of the closed GOP is I P B B P B B. The structure of the open GOP is I B B P B B. If the functions of resuming or auto splitting a video are enabled, the value is always false, i.e., open GOP.)
UINT8	<b>VFR</b>	The factor of the variable frame rate (For example, 2 and 4 mean that their frame rate are 1/2 and 1/4 respectively. (e.g., VFR = 2, 60P->30P))
UINT8	<b>ColorStyle</b>	The color space of the video (The value 0 is for TV. The value 1 is for PC.)
UINT8	<b>EntropyMode</b>	The entropy mode of the H.264 bitstream (If the mode is CAVLC, the value is 0. If the mode is CABAC, the value is 1.)
BOOL8	<b>IsVFR</b>	The flag indicating if a track is variable frame rate (If the value is TRUE and this is a VFR track. If the value is FALSE, this is a Fix frame rate track.)

Table 5-12. Definition of **AMP\_VIDEO\_TRACK\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

### 5.2.2.4 AmpFormat\_CopyMovieInfo > AMP\_AUDIO\_TRACK\_INFO\_s

Type	Field	Description
UINT32	<b>SampleRate</b>	The sample rate (Hz) of the audio track

Type	Field	Description
BOOL8	<b>IsDefault</b>	The flag indicating the track is the default audio track
UINT8	<b>Channels</b>	The number of channels in the audio track
UINT8	<b>BitsPerSample</b>	The bits per sample of the audio track (e.g., 8 bits and 16 bits)
UINT8	<b>Endian</b>	The endian type of the audio track (e.g., Big endian or little endian)

Table 5-13. Definition of **AMP\_AUDIO\_TRACK\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

#### 5.2.2.5 AmpFormat\_CopyMovieInfo > AMP\_TEXT\_TRACK\_INFO\_s

Type	Field	Description
BOOL8	<b>IsDefault</b>	The flag indicating the track is the default text track

Table 5-14. Definition of **AMP\_TEXT\_TRACK\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

#### 5.2.2.6 AmpFormat\_CopyMovieInfo > AMP\_MUX\_PROC\_INFO\_s

Type	Field	Description
UINT64 [AMP_FORMAT_MAX_TRACK_PER_MEDIA]	<b>ResumeDTS</b>	The DTS of the last resuming

Table 5-15. Definition of **AMP\_MUX\_PROC\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

#### 5.2.2.7 AmpFormat\_CopyMovieInfo > AMP\_DMUX\_PROC\_INFO\_s

Type	Field	Description
AMP_DMUX_TRACK_PROC_INFO_s [AMP_FORMAT_MAX_TRACK_PER_MEDIA]	<b>Track</b>	The demuxing information of each track ( <a href="#">Section 5.2.2.8</a> )

Table 5-16. Definition of **AMP\_DMUX\_PROC\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

### 5.2.2.8 AmpFormat\_CopyMovieInfo > AMP\_DMX\_TRACK\_PROC\_INFO\_s

Type	Field	Description
UINT64 [AMP_FORMAT_MAX_TRACK_PER_MEDIA]	<b>EndPTS</b>	The largest PTS of the frame that has been fed to FIFO

Table 5-17. Definition of **AMP\_DMX\_TRACK\_PROC\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

### 5.2.2.9 AmpFormat\_CopyMovieInfo > AMP\_ISO\_PRIV\_INFO\_s

Type	Field	Description
UINT32	<b>CreateTime</b>	Creation time of the media
UINT32	<b>ModifyTime</b>	Modification time of the media
UINT32	<b>FtypSize</b>	The box size of the Iso ftyp
UINT32	<b>TrickRecDivisor</b>	The divisor factor for HFR
ISO_TRACK_INFO_s [AMP_FORMAT_MAX_TRACK_PER_MEDIA]	<b>TrackInfo</b>	The track information in the private data ( <a href="#">Section 5.2.2.10</a> )
UINT64	<b>FrameDataSize</b>	The mdat size used in editor
UINT64	<b>FrameDataPos</b>	The mdat position used in editor
UINT64	<b>MaxCachedSize</b>	Max cached data size of a stream
BOOL8	<b>EnableCO64</b>	The flag to enable large mdat size

Table 5-18. Definition of **AMP\_ISO\_PRIV\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

### 5.2.2.10 AmpFormat\_CopyMovieInfo > ISO\_TRACK\_INFO\_s

Type	Field	Description
union { ISO_VIDEO_TRACK_INFO_s Video; ISO_AUDIO_TRACK_INFO_s Audio; ISO_TEXT_TRACK_INFO_s Text; }	<b>Info</b>	Video: Video track information ( <a href="#">Section 5.2.2.11</a> ) Audio: Audio track information ( <a href="#">Section 5.2.2.12</a> ) Text: Text track information ( <a href="#">Section 5.2.2.13</a> )

Table 5-19. Definition of **ISO\_TRACK\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

### 5.2.2.11 AmpFormat\_CopyMovieInfo > ISO\_VIDEO\_TRACK\_INFO\_s

Type	Field	Description
UINT8 [AMP_FORMAT_MAX_SPS_LENGTH]	<b>SPS</b>	The SPS of H264
UINT8 [AMP_FORMAT_MAX_PPS_LENGTH]	<b>PPS</b>	The PPS of H264
UINT32	<b>KeyFrameNo</b>	The number of the key frame in current index buffer
UINT32	<b>KeyFrameCount</b>	The count of the key frame in current index buffer
UINT32	<b>SttsCount</b>	The count of the STTS entry in current index buffer
UINT32	<b>FrameNumGOP</b>	The frame number counter in a GOP (Reset in Idr)
UINT64	<b>TmpV</b>	The start offset of the index (It stores the size entry of the video track)
UINT64	<b>TmpVo</b>	The start offset of the index (It stores the file offset entry of the video track)
UINT64	<b>TmpK</b>	The start offset of the index (It stores the key frame entry of the video track)
UINT64	<b>TmpCtts</b>	The start offset of the index (It stores the CTTS entry of the video track)
UINT64	<b>TmpStts</b>	The start offset of the index (It stores the STTS entry of the video track)
UINT16	<b>SPSLen</b>	The SPS size of H264
UINT16	<b>PPSLen</b>	The PPS size of H264

Table 5-20. Definition of **ISO\_VIDEO\_TRACK\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

### 5.2.2.12 AmpFormat\_CopyMovieInfo > ISO\_AUDIO\_TRACK\_INFO\_s

Type	Field	Description
UINT32	<b>SttsCount</b>	The count of the STTS entry in current index buffer
UINT64	<b>TmpA</b>	The start offset of the index (It stores the size entry of the audio track)
UINT64	<b>TmpAo</b>	The start offset of the index (It stores the file offset entry of the audio track)
UINT64	<b>TmpStts</b>	The start offset of the index (It stores the STTS entry of the audio track)

Table 5-21. Definition of *ISO\_AUDIO\_TRACK\_INFO\_s* for Format SDK6 API *AmpFormat\_CopyMovieInfo()*.

### 5.2.2.13 AmpFormat\_CopyMovieInfo > ISO\_TEXT\_TRACK\_INFO\_s

Type	Field	Description
UINT32	<b>SttsCount</b>	The count of the STTS entry in the current index buffer
UINT64	<b>TmpT</b>	The start offset of the index (It stores the size entry of the text track)
UINT64	<b>TmpTo</b>	The start offset of the index (It stores the file offset entry of the text track)
UINT64	<b>TmpStts</b>	The start offset of the index (It stores the STTS entry of the text track)

Table 5-22. Definition of *ISO\_TEXT\_TRACK\_INFO\_s* for Format SDK6 API *AmpFormat\_CopyMovieInfo()*.

### 5.2.2.14 AmpFormat\_CopyMovieInfo > AMP\_MATROSKA\_PRIV\_INFO\_s

Type	Field	Description
UINT64	<b>SegmentStart</b>	The start position of Segment content, right behind tag and size
UINT64	<b>InfoPos</b>	The position of Info
UINT64	<b>TracksPos</b>	The position of Tracks
UINT64	<b>CuesPos</b>	The position of Cues
UINT32	<b>TimecodeScale</b>	Timestamp scale in nanoseconds
INT64	<b>DateUtc</b>	Date of the origin of timestamp
UINT8[16]	<b>SegmentUid</b>	A randomly generated unique ID to identify the current segment between many others (128 bits)
MATROSKA_TRACK_INFO_s [AMP_FORMAT_MAX_TRACK_PER_MEDIA]	<b>TrackInfo</b>	Tracks ( <a href="#">Section 5.2.2.15</a> )

Table 5-23. Definition of *AMP\_MATROSKA\_PRIV\_INFO\_s* for Format SDK6 API *AmpFormat\_CopyMovieInfo()*.

### 5.2.2.15 AmpFormat\_CopyMovieInfo > MATROSKA\_TRACK\_INFO\_s

Type	Field	Description
UINT32	<b>TrackUid</b>	Track UID
union { MATROSKA_VIDEO_TRACK_INFO_s Video; MATROSKA_AUDIO_TRACK_INFO_s Audio; MATROSKA_TEXT_TRACK_INFO_s Text; }	<b>Info</b>	Video: The information of the video track ( <a href="#">Section 5.2.2.16</a> )  Audio: The information of the audio track ( <a href="#">Section 5.2.2.17</a> )  Text: The information of the text track ( <a href="#">Section 5.2.2.18</a> )

Table 5-24. Definition of **MATROSKA\_TRACK\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

### 5.2.2.16 AmpFormat\_CopyMovieInfo > MATROSKA\_VIDEO\_TRACK\_INFO\_s

Type	Field	Description
UINT8 [AMP_FORMAT_MAX_SPS_LENGTH]	<b>SPS</b>	SPS of the H264 bitstream
UINT8 [AMP_FORMAT_MAX_PPS_LENGTH]	<b>PPS</b>	PPS of the H264 bitstream
UINT16	<b>SPSLen</b>	The SPS size of the H264 bitstream
UINT16	<b>PPSLen</b>	The PPS size of the H264 bitstream

Table 5-25. Definition of **MATROSKA\_VIDEO\_TRACK\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

### 5.2.2.17 AmpFormat\_CopyMovieInfo > MATROSKA\_AUDIO\_TRACK\_INFO\_s

Type	Field	Description
UINT8	<b>Profile</b>	AAC profile
UINT8	<b>Channels</b>	The number of channels kept in codec private
UINT32	<b>SamplingFrequency</b>	The sampling frequency kept in codec private

Table 5-26. Definition of **MATROSKA\_AUDIO\_TRACK\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

### 5.2.2.18 AmpFormat\_CopyMovieInfo > MATROSKA\_TEXT\_TRACK\_INFO\_s

Type	Field	Description
UINT8[4]	<b>Resv</b>	Reserved

Table 5-27. Definition of **MATROSKA\_TEXT\_TRACK\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

### 5.2.2.19 AmpFormat\_CopyMovieInfo > AMP\_THUMB\_FRAME\_INFO\_s

Type	Field	Description
UINT32	<b>SeqNum</b>	The sequence number
UINT32	<b>Type</b>	The image type
UINT32	<b>Pos</b>	The image position
UINT32	<b>Size</b>	The image size
UINT16	<b>Width</b>	The image width
UINT16	<b>Height</b>	The image height

Table 5-28. Definition of **AMP\_THUMB\_FRAME\_INFO\_s** for Format SDK6 API **AmpFormat\_CopyMovieInfo()**.

Confidential  
For HAOTEK Only

### 5.2.3 AmpFormat\_CopySoundInfo

#### API Syntax:

**AmpFormat\_CopySoundInfo** (AMP\_SOUND\_INFO\_s \* dstInfo, AMP\_SOUND\_INFO\_s \* srcInfo)

#### Function Description:

- This function is used to copy a sound information object to a destination object.

#### Parameters:

Type	Parameter	Description
AMP_SOUND_INFO_s *	<b>dstInfo</b>	The destination sound information object. ( <b>AMP_SOUND_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.2.3.1</a> below for more details.
AMP_SOUND_INFO_s *	<b>srcInfo</b>	The source sound information object. ( <b>AMP_SOUND_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.2.3.1</a> below for more details.

Table 5-29. Parameters for Format SDK6 API **AmpFormat\_CopySoundInfo()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-30. Returns for Format SDK6 API **AmpFormat\_CopySoundInfo()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



### 5.2.3.1 AmpFormat\_CopySoundInfo > AMP\_SOUND\_INFO\_s

Type	Field	Description
UINT8	<b>MediaType</b>	The media type (Please refer to AMP_MEDIA_INFO_TYPE_e)
BOOL8	<b>Valid</b>	The value indicating if the media context is valid.
UINT8	<b>SubFormat</b>	The sub_format of the media (MSMP4, FUJIMOV)
UINT32	<b>MagicPattern</b>	0x12345678 (used for recovery)
UINT64	<b>Size</b>	The file size
char [MAX_FILENAME_LENGTH]	<b>Name</b>	The media name
char [AMP_FORMAT_MAX_DATE_SIZE]	<b>Date</b>	The creation date of the media
char [AMP_FORMAT_MAX_TIME_SIZE]	<b>Time</b>	The creation time of the media
AMP_MEDIA_TRACK_INFO_s [AMP_FORMAT_MAX_TRACK_PER_MEDIA]	<b>Track</b>	The tracks in a sound file (Please refer to <a href="#">Section 5.2.2.2</a> )
union { AMP_EXT_PRIV_INFO_s Ext; AMP_ISO_PRIV_INFO_s Iso; AMP_MATROSKA_PRIV_INFO_s Matroska; }	<b>PrivInfo</b>	Ext: The private data for the information of Ext Format ( <a href="#">Section 5.2.1.4</a> ) Iso: The private data for the information of Iso Format ( <a href="#">Section 5.2.2.9</a> ) Matroska: The private data for the information of Matroska Format ( <a href="#">Section 5.2.2.14</a> )
UINT8	<b>TrackCount</b>	The number of tracks

Table 5-31. Definition of **AMP\_SOUND\_INFO\_s** for Format SDK6 API **AmpFormat\_CopySoundInfo()**.

## 5.2.4 AmpFormat\_FlushMediaInfo

### API Syntax:

**AmpFormat\_FlushMediaInfo** (void)

### Function Description:

- This function is used to flush all of the media information objects in the Format module.

### Parameters:

None

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-32. Returns for Format SDK6 API **AmpFormat\_FlushMediaInfo()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.2.5 AmpFormat\_GetImageInfo

### API Syntax:

**AmpFormat\_GetImageInfo** (char \* name, AMP\_DMx\_FORMAT\_PARSE\_FP parse, AMP\_STREAM\_HDLR\_s \* stream, AMP\_IMAGE\_INFO\_s \*\* info)

### Function Description:

- This function is used to get an image information object and parse it if it is not loaded (used by Demuxer).

### Parameters:

Type	Parameter	Description
char *	<b>name</b>	The name of the image file
AMP_DMx_FORMAT_PARSE_FP	<b>parse</b>	The function to parse the image information object
AMP_STREAM_HDLR_s *	<b>stream</b>	The stream to access the image file. ( <b>AMP_STREAM_HDLR_s</b> is defined in <code>Stream.h</code> ) Please refer to Section 8.2.1.2 for more details.
AMP_IMAGE_INFO_s **	<b>info</b>	The returned image information object. ( <b>AMP_IMAGE_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.2.1.1</a> for more details.

Table 5-33. Parameters for Format SDK6 API **AmpFormat\_GetImageInfo()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-34. Returns for Format SDK6 API **AmpFormat\_GetImageInfo()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 5.2.6 AmpFormat\_GetInitDefaultCfg

### API Syntax:

**AmpFormat\_GetInitDefaultCfg** (AMP\_FORMAT\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the Format module.

### Parameters:

Type	Parameter	Description
AMP_FORMAT_INIT_CFG_s *	<b>config</b>	The returned configuration ( <b>AMP_FORMAT_INIT_CFG_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.2.6.1</a> for more details.

Table 5-35. Parameters for Format SDK6 API **AmpFormat\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-36. Returns for Format SDK6 API **AmpFormat\_GetInitDefaultCfg()**.

### Example

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.2.6.1 AmpFormat\_GetInitDefaultCfg > AMP\_FORMAT\_INIT\_CFG\_s

Type	Field	Description
UINT32	<b>MaxMovie</b>	The maximum number of Movie information objects in the Format module
UINT32	<b>MaxImage</b>	The maximum number of Image information objects in the Format module
UINT32	<b>MaxSound</b>	The maximum number of Sound information objects in the Format module
UINT8 *	<b>Buffer</b>	The work buffer of the Format module
UINT32	<b>BufferSize</b>	The work buffer size
UINT8	<b>AmbaMainVer</b>	The main version of AMBA Box
UINT8	<b>AmbaSubVer</b>	The sub version of AMBA Box

Table 5-37. Definition of **AMP\_FORMAT\_INIT\_CFG\_s** for Format SDK6 API **AmpFormat\_GetInitDefaultCfg()**.

## 5.2.7 AmpFormat\_GetMovieInfo

### API Syntax:

**AmpFormat\_GetMovieInfo** (char \* name, AMP\_DMX\_FORMAT\_PARSE\_FP parse, AMP\_STREAM\_HDLR\_s \* stream, AMP\_MOVIE\_INFO\_s \*\* info)

### Function Description:

- This function is used to get a movie information object and parse it if it is not loaded (used by Demuxer).

### Parameters:

Type	Parameter	Description
char *	<b>name</b>	The name of a movie file
AMP_DMX_FORMAT_PARSE_FP	<b>parse</b>	The function to parse the Movie information object
AMP_STREAM_HDLR_s *	<b>stream</b>	The stream to access the movie file. ( <b>AMP_STREAM_HDLR_s</b> is defined in <code>Stream.h</code> ) Please refer to <a href="#">Section 8.2.1.2</a> for more details.
AMP_MOVIE_INFO_s **	<b>info</b>	The returned movie information object. Please refer to <a href="#">Section 5.2.2.1</a> for more details.

Table 5-38. Parameters for Format SDK6 API **AmpFormat\_GetMovieInfo()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-39. Returns for Format SDK6 API **AmpFormat\_GetMovieInfo()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 5.2.8 AmpFormat\_GetRequiredBufferSize

### API Syntax:

**AmpFormat\_GetRequiredBufferSize** (UINT32 maxMovie, UINT32 maxImage, UINT32 maxSound)

### Function Description:

- This function is used to get the required buffer size for initializing the Format module.

### Parameters:

Type	Parameter	Description
UINT32	<b>maxMovie</b>	The maximum number of Movie information objects in the Format module
UINT32	<b>maxImage</b>	The maximum number of Image information objects in the Format module
UINT32	<b>maxSound</b>	The maximum number of Sound information objects in the Format module

Table 5-40. Parameters for Format SDK6 API **AmpFormat\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 5-41. Returns for Format SDK6 API **AmpFormat\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 5.2.9 AmpFormat\_GetSoundInfo

### API Syntax:

**AmpFormat\_GetSoundInfo** (char \* name, AMP\_DMXX\_FORMAT\_PARSE\_FP parse, AMP\_STREAM\_HDLR\_s \* stream, AMP\_SOUND\_INFO\_s \*\* info)

### Function Description:

- This function is used to get a sound information object and parse it if it is not loaded (used by De-muxer).

### Parameters:

Type	Parameter	Description
char *	<b>name</b>	The name of a sound file
AMP_DMXX_FORMAT_PARSE_FP	<b>parse</b>	The function to parse the Sound information object
AMP_STREAM_HDLR_s *	<b>stream</b>	The stream to access the sound file. ( <b>AMP_STREAM_HDLR_s</b> is defined in <code>Stream.h</code> ) Please refer to Section 8.2.1.2 for more details.
AMP_SOUND_INFO_s **	<b>info</b>	The returned sound information object. Please refer to <a href="#">Section 5.2.3.1</a> for more details.

Table 5-42. Parameters for Format SDK6 API **AmpFormat\_GetSoundInfo()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-43. Returns for Format SDK6 API **AmpFormat\_GetSoundInfo()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 5.2.10 AmpFormat\_Init

### API Syntax:

**AmpFormat\_Init** (AMP\_FORMAT\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initiate the Format module.

### Parameters:

Type	Parameter	Description
AMP_FORMAT_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the Format module. ( <b>AMP_FORMAT_INIT_CFG_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.2.6.1</a> for more details.

Table 5-44. Parameters for Format SDK6 API **AmpFormat\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-45. Returns for Format SDK6 API **AmpFormat\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 5.2.11 AmpFormat\_NewImageInfo

### API Syntax:

**AmpFormat\_NewImageInfo** (char \* name, AMP\_IMAGE\_INFO\_s \*\* info)

### Function Description:

- This function is used to new an image information object.

### Parameters:

Type	Parameter	Description
char *	<b>name</b>	The name of an image file
AMP_IMAGE_INFO_s **	<b>info</b>	The returned image information object. ( <b>AMP_IMAGE_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.2.1.1</a> for more details.

Table 5-46. Parameters for Format SDK6 API **AmpFormat\_NewImageInfo()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-47. Returns for Format SDK6 API **AmpFormat\_NewImageInfo()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 5.2.12 AmpFormat\_NewMovieInfo

### API Syntax:

**AmpFormat\_NewMovieInfo** (char \* name, AMP\_MOVIE\_INFO\_s \*\* info)

### Function Description:

- This function is used to new a Movie information object (used by Muxer).

### Parameters:

Type	Parameter	Description
char *	<b>name</b>	The name of a movie file
AMP_MOVIE_INFO_s **	<b>info</b>	The returned Movie information object. ( <b>AMP_MOVIE_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.2.2.1</a> for more details.

Table 5-48. Parameters for Format SDK6 API **AmpFormat\_NewMovieInfo()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e Please refer to <a href="#">Chapter 10</a> .

Table 5-49. Returns for Format SDK6 API **AmpFormat\_NewMovieInfo()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

### 5.2.13 AmpFormat\_NewSoundInfo

#### API Syntax:

**AmpFormat\_NewSoundInfo** (char \* name, AMP\_SOUND\_INFO\_s \*\* info)

#### Function Description:

- This function is used to new a Sound information object.

#### Parameters:

Type	Parameter	Description
char *	<b>name</b>	The name of a sound file
AMP_SOUND_INFO_s **	<b>info</b>	The returned Sound information object. ( <b>AMP_SOUND_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.2.3.1</a> for more details.

Table 5-50. Parameters for Format SDK6 API **AmpFormat\_NewSoundInfo()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-51. Returns for Format SDK6 API **AmpFormat\_NewSoundInfo()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

None

## 5.2.14 AmpFormat\_RellmageInfo

### API Syntax:

**AmpFormat\_RellmageInfo** (AMP\_IMAGE\_INFO\_s \* info, BOOL remove)

### Function Description:

- This function is used to release an image information object (unlock the media).

### Parameters:

Type	Parameter	Description
AMP_IMAGE_INFO_s *	<b>info</b>	The image information object being released. ( <b>AMP_IMAGE_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.2.1.1</a> for more details.
BOOL	<b>remove</b>	Remove the Image information object from the Format module after it is closed

Table 5-52. Parameters for Format SDK6 API **AmpFormat\_RellmageInfo()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-53. Returns for Format SDK6 API **AmpFormat\_RellmageInfo()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.2.15 AmpFormat\_RelMovieInfo

### API Syntax:

**AmpFormat\_RelMovieInfo** (AMP\_MOVIE\_INFO\_s \* info, BOOL remove)

### Function Description:

- This function is used to release a movie information object (unlock the media).

### Parameters:

Type	Parameter	Description
AMP_MOVIE_INFO_s *	<b>info</b>	A movie information object being released. ( <b>AMP_MOVIE_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.2.2.1</a> for more details.
BOOL	<b>remove</b>	Remove the Movie information object from the Format module after it is closed.

Table 5-54. Parameters for Format SDK6 API **AmpFormat\_RelMovieInfo()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-55. Returns for Format SDK6 API **AmpFormat\_RelMovieInfo()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.2.16 AmpFormat\_RelSoundInfo

### API Syntax:

**AmpFormat\_RelSoundInfo** (AMP\_SOUND\_INFO\_s \* info, BOOL remove)

### Function Description:

- The function is used to release a sound information object (unlock the media).

### Parameters:

Type	Parameter	Description
AMP_SOUND_INFO_s *	<b>info</b>	The sound information being released. ( <b>AMP_SOUND_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.2.3.1</a> for more details.
BOOL	<b>remove</b>	Remove the Sound information object from the Format module after it is closed

Table 5-56. Parameters for Format SDK6 API **AmpFormat\_RelSoundInfo()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-57. Returns for Format SDK6 API **AmpFormat\_RelSoundInfo()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#).

## 5.3 Dummy

This section describes the Sample format flow implementation. DummyMux is a muxing format without any container information, i.e., raw data only.

Confidential  
For HAOTEK Only

### 5.3.1 AmpDummyMux\_Create

#### API Syntax:

**AmpDummyMux\_Create** (AMP\_DUMMY\_MUX\_CFG\_s \* config, AMP\_MUX\_FORMAT\_HDLR\_s \*\* hdlr)

#### Function Description:

- This function is used to create a DummyMux handler.

#### Parameters:

Type	Parameter	Description
AMP_DUMMY_MUX_CFG_s *	<b>config</b>	The configuration used to create a DummyMux handler. ( <b>AMP_DUMMY_MUX_CFG_s</b> is defined in <code>DummyMux.h</code> ) Please refer to <a href="#">Section 5.3.1.1</a> below for more details.
AMP_MUX_FORMAT_HDLR_s **	<b>hdlr</b>	The returned DummyMux handler. Please refer to <a href="#">Section 5.3.2.1</a> for definition.

Table 5-58. Parameters for Dummy SDK6 API **AmpDummyMux\_Create()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-59. Returns for Dummy SDK6 API **AmpDummyMux\_Create()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



### 5.3.1.1 AmpDummyMux\_Create > AMP\_DUMMY\_MUX\_CFG\_s

Type	Field	Description
AMP_STREAM_HDLR_s *	<b>Stream</b>	Stream handler
union {AMP_MUX_PARAM_MOVIE_s Movie; AMP_MUX_PARAM_SOUND_s Sound; AMP_MUX_PARAM_IMAGE_s Image}	<b>Param</b>	Movie: The muxing parameters of a movie. Please refer to <a href="#">Section 5.3.1.2</a> below for more details. Sound: The muxing parameters of a sound. Please refer to <a href="#">Section 5.3.1.3</a> below for more details. Image: The muxing parameters of an image. Please refer to <a href="#">Section 5.3.1.4</a> below for more details.
UINT8	<b>MediaType</b>	Media type (Please refer to AMP_MEDIA_TRACK_TYPE_e.)

Table 5-60. Definition of **AMP\_DUMMY\_MUX\_CFG\_s** for Dummy SDK6 API **AmpDummyMux\_Create()**.

### 5.3.1.2 AmpDummyMux\_Create > AMP\_MUX\_PARAM\_MOVIE\_s

Type	Field	Description
UINT32	<b>MaxDuration</b>	The maximum duration of a movie file
UINT64	<b>MaxSize</b>	The maximum size of a movie file

Table 5-61. Definition of **AMP\_MUX\_PARAM\_MOVIE\_s** for Dummy SDK6 API **AmpDummyMux\_Create()**.

### 5.3.1.3 AmpDummyMux\_Create > AMP\_MUX\_PARAM\_SOUND\_s

Type	Field	Description
UINT32	<b>MaxDuration</b>	The maximum duration of sound file
UINT64	<b>MaxSize</b>	The maximum size of a sound file

Table 5-62. Definition of **AMP\_MUX\_PARAM\_SOUND\_s** for Dummy SDK6 API **AmpDummyMux\_Create()**.

### 5.3.1.4 AmpDummyMux\_Create > AMP\_MUX\_PARAM\_IMAGE\_s

Type	Field	Description
UINT8[4]	<b>Resv</b>	Reserved

Table 5-63. Definition of **AMP\_MUX\_PARAM\_IMAGE\_s** for Dummy SDK6 API **AmpDummyMux\_Create()**.

## 5.3.2 AmpDummyMux\_Delete

### API Syntax:

**AmpDummyMux\_Delete** (AMP\_MUX\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete a DummyMux handler.

### Parameters:

Type	Parameter	Description
AMP_MUX_FORMAT_HDLR_s **	hdlr	The DummyMux handler being deleted. ( <b>AMP_MUX_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.3.2.1</a> below for more details.

Table 5-64. Parameters for Dummy SDK6 API **AmpDummyMux\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-65. Returns for Dummy SDK6 API **AmpDummyMux\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.3.2.1 AmpDummyMux\_Delete > AMP\_MUX\_FORMAT\_HDLR\_s

Type	Field	Description
AMP_MUX_FORMAT_s *	<b>Func</b>	The interface of Muxing Format (Please refer to AMP_MUX_FORMAT_s)
AMP_MEDIA_INFO_s *	<b>Media</b>	Media information object
AMP_STREAM_HDLR_s *	<b>Stream</b>	Stream handler
union {AMP_MUX_PARAM_MOVIE_s Movie; AMP_MUX_PARAM_SOUND_s Sound; AMP_MUX_PARAM_IMAGE_s Image}	<b>Param</b>	Movie: The muxing parameters of a Movie information object. Please refer to <a href="#">Section 5.3.1.2</a> for more details.
		The muxing parameters of a Sound information object. Please refer to <a href="#">Section 5.3.1.3</a> for more details.
		The muxing parameters of a Image information object. Please refer to <a href="#">Section 5.3.1.4</a> for more details.

Table 5-66. Definition of AMP\_MUX\_FORMAT\_HDLR\_s for Dummy SDK6 API AmpDummyMux\_Delete().

### 5.3.3 AmpDummyMux\_GetDefaultCfg

#### API Syntax:

**AmpDummyMux\_GetDefaultCfg** (UINT8 mediaType, AMP\_DUMMY\_MUX\_CFG\_s \* config)

#### Function Description:

- This function is used to get the default configuration of a DummyMux handler.

#### Parameters:

Type	Parameter	Description
UINT8	<b>mediaType</b>	Media type (Please refer to <b>AMP_MEDIA_INFO_TYPE_e</b> )
AMP_DUMMY_MUX_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_DUMMY_MUX_CFG_s</b> is defined in <code>DummyMux.h</code> ) Please refer to <a href="#">Section 5.3.1.1</a> for more details.

Table 5-67. Parameters for Dummy SDK6 API **AmpDummyMux\_GetDefaultCfg()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-68. Returns for Dummy SDK6 API **AmpDummyMux\_GetDefaultCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.3.4 AmpDummyMux\_GetInitDefaultCfg

#### API Syntax:

**AmpDummyMux\_GetInitDefaultCfg** (AMP\_DUMMY\_MUX\_INIT\_CFG\_s \* config)

#### Function Description:

- This function is used to get the default configuration of the DummyMux module.

#### Parameters:

Type	Parameter	Description
AMP_DUMMY_MUX_INIT_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_DUMMY_MUX_INIT_CFG_s</b> is defined in <code>DummyMux.h</code> ) Please refer to <a href="#">Section 5.3.4.1</a> for more details.

Table 5-69. Parameters for Dummy SDK6 API **AmpDummyMux\_GetInitDefaultCfg()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-70. Returns for Dummy SDK6 API **AmpDummyMux\_GetInitDefaultCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

#### 5.3.4.1 AmpDummyMux\_Init > AMP\_DUMMY\_MUX\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the DummyMux module
UINT32	<b>BufferSize</b>	The size of the work buffer
UINT8	<b>MaxHdlr</b>	The maximum number of DummyMux handlers

Table 5-71. Definition of **AMP\_DUMMY\_MUX\_INIT\_CFG\_s** for Dummy SDK6 API **AmpDummyMux\_Init()**.

### 5.3.5 AmpDummyMux\_GetRequiredBufferSize

#### API Syntax:

**AmpDummyMux\_GetRequiredBufferSize** (UINT8 maxHdlr)

#### Function Description:

- This function is used to get the required buffer size of the DummyMux module.

#### Parameters:

Type	Parameter	Description
UINT8	maxHdlr	The maximum number of DummyMux handlers

Table 5-72. Parameters for Dummy SDK6 API **AmpDummyMux\_GetRequiredBufferSize()**.

#### Returns:

Return	Description
The required buffer size	The required buffer size

Table 5-73. Returns for Dummy SDK6 API **AmpDummyMux\_GetRequiredBufferSize()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

None

### 5.3.6 AmpDummyMux\_Init

#### API Syntax:

**AmpDummyMux\_Init** (AMP\_DUMMY\_MUX\_INIT\_CFG\_s \* config)

#### Function Description:

- This function is used to initialize the DummyMux module.

#### Parameters:

Type	Parameter	Description
AMP_DUMMY_MUX_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the module. ( <b>AMP_DUMMY_MUX_INIT_CFG_s</b> is defined in <code>DummyMux.h</code> ) Please refer to <a href="#">Section 5.3.4.1</a> for more details.

Table 5-74. Parameters for Dummy SDK6 API **AmpDummyMux\_Init()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-75. Returns for Dummy SDK6 API **AmpDummyMux\_Init()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.4 EXIF

This section describes the APIs of EXIF Muxing/Demuxing Format module. Users can use ExifMux to add data into the picture and use ExifDmx to parse data from a picture.

The ExifMux/ExifDmx module include below function implementation:

1. Initialize ExifMux/ExifDmx
2. Create ExifMux/ExifDmx handler
3. Delete ExifMux/ExifDmx handler

Confidential  
For HAOTEK Only



## 5.4.1 AmpExifDmx\_Create

### API Syntax:

**AmpExifDmx\_Create** (AMP\_EXIF\_DMX\_CFG\_s \* config, AMP\_DMX\_FORMAT\_HDLR\_s \*\* hdlr )

### Function Description:

- This function is used to create an ExifDmx handler.

### Parameters:

Type	Parameter	Description
AMP_EXIF_DMX_CFG_s *	<b>config</b>	The configuration used to create an ExifDmx handler. ( <b>AMP_EXIF_DMX_CFG_s</b> is defined in <code>ExifDmx.h</code> ) Please refer to <a href="#">Section 5.4.1.1</a> below for definition.
AMP_DMX_FORMAT_HDLR_s **	<b>hdlr</b>	The returned ExifDmx handler. Please refer to <a href="#">Section 5.4.1.2</a> below for more details.

Table 5-76. Parameters for EXIF SDK6 API **AmpExifDmx\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-77. Returns for EXIF SDK6 API **AmpExifDmx\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.4.1.1 AmpExifDmx\_Create > AMP\_EXIF\_DMX\_CFG\_s

Type	Field	Description
AMP_STREAM_HDLR_s *	<b>Stream</b>	Stream handler (Please refer to <a href="#">Section 8.2.1.2</a> )

Table 5-78. Definition of **AMP\_EXIF\_DMX\_CFG\_s** for EXIF SDK6 API **AmpExifDmx\_Create()**.

#### 5.4.1.2 Amp\_ExifDmx\_Create > AMP\_DMX\_FORMAT\_HDLR\_s

Type	Field	Description
AMP_DMX_FORMAT_s*	<b>Func</b>	The interface of Demuxing Format (Please refer to <a href="#">Section 5.4.1.3</a> )
AMP_MEDIA_INFO_s*	<b>Media</b>	Media information object (AMP_MEDIA_INFO_s is defined in <code>FormatDef.h</code> , refer to <a href="#">Section 5.4.1.4</a> )
AMP_STREAM_HDLR_s*	<b>Stream</b>	Stream handler (AMP_STREAM_HDLR_s is defined in <code>Stream.h</code> , please refer to <a href="#">Section 8.2.1.2</a> )
union { AMP_DMX_PARAM_MOVIE_s Movie; AMP_DMX_PARAM_SOUND_s Sound; AMP_DMX_PARAM_IMAGE_s Image; }	<b>Param</b>	Movie: The demuxing parameters of a Movie information object ( <a href="#">Section 5.4.1.5</a> ) Sound: The demuxing parameters of a Sound information object ( <a href="#">Section 5.4.1.6</a> ) Image: The demuxing parameters of an Image information object ( <a href="#">Section 5.4.1.7</a> )

Table 5-79. Definition of **AMP\_DMX\_FORMAT\_HDLR\_s** for Exif SDK6 API **AmpExifDmx\_Create()**.

#### 5.4.1.3 Amp\_ExifDmx\_Create > AMP\_DMX\_FORMAT\_s

Type	Field	Description
int(*) (AMP_DMX_FORMAT_HDLR_s *)	<b>Open</b>	The interface to open a Format handler (Handler)
int(*) (AMP_DMX_FORMAT_HDLR_s *)	<b>Close</b>	The interface to close a Format handler (Handler)
int(*) (AMP_DMX_FORMAT_HDLR_s *, UINT64)	<b>Process</b>	The interface to perform the demuxing of a media (Handler, The data unit to demux)
UINT64(*) (AMP_DMX_FORMAT_HDLR_s *)	<b>GetMediaTime</b>	The interface to get the current media time (ms) (Handler)
int(*) (AMP_DMX_FORMAT_HDLR_s *, UINT64, UINT32, UINT32)	<b>Seek</b>	The interface to seek a specified time offset (ms) and continue with the specified direction and speed (Handler, Target time, Direction, Speed)
int(*) (AMP_DMX_FORMAT_HDLR_s *, UINT8, UINT64, UINT8)	<b>FeedFrame</b>	The interface to feed a specified frame to FIFO (Handler, Track Id, Target time, Frame type)

Type	Field	Description
int(*) (AMP_DMX_FORMAT_HDLR_s *, UINT32, UINT32)	<b>Func</b>	The interface to execute special commands (Handler, Parameter1, Parameter2)

Table 5-80. Definition of **AMP\_DMX\_FORMAT\_s** for Exif SDK6 API **AmpExifDmx\_Create()**.

#### 5.4.1.4 Amp\_ExifDmx\_Create > AMP\_MEDIA\_INFO\_s

Type	Field	Description
UINT8	<b>MediaType</b>	The media type (Please refer to AMP MEDIA INFO TYPE e)
BOOL8	<b>Valid</b>	The value indicating if the media context is valid
UINT8	<b>SubFormat</b>	The sub format of the media (MSMP4, FUJIMOV)
UINT32	<b>MagicPattern</b>	The Magic pattern 0x12345678 (used for recovery)
UINT64	<b>Size</b>	The file size
char [MAX_FILENAME_LENGTH]	<b>Name</b>	The media name
char [AMP_FORMAT_MAX_DATE_SIZE]	<b>Date</b>	The creation date of the media
char [AMP_FORMAT_MAX_TIME_SIZE]	<b>Time</b>	The creation time of the media

Table 5-81. Definition of **AMP\_MEDIA\_INFO\_s** for EXIF SDK6 API **AmpExifDmx\_Create()**.

#### 5.4.1.5 Amp\_ExifDmx\_Create > AMP\_DMX\_PARAM\_MOVIE\_s

Type	Field	Description
UINT8	<b>Direction</b>	The direction in demuxing
UINT8	<b>Speed</b>	Demuxing speed (e.q., 1, 2, 4, 8, and 16)
BOOL8	<b>End</b>	Notify EOS once the last frame is fed

Table 5-82. Definition of **AMP\_DMX\_PARAM\_MOVIE\_s** for EXIF SDK6 API **AmpExifDmx\_Create()**.

#### 5.4.1.6 Amp\_ExifDmx\_Create > AMP\_DMX\_PARAM\_SOUND\_s

Type	Field	Description
UINT8	<b>Direction</b>	The direction in demuxing
UINT8	<b>Speed</b>	Demuxing speed (e.q., 1, 2, 4, 8, and 16)

Table 5-83. Definition of **AMP\_DMX\_PARAM\_SOUND\_s** for EXIF SDK6 API **AmpExifDmx\_Create()**.

5.4.1.7 Amp\_ExifDmx\_Create > AMP\_DMX\_PARAM\_IMAGE\_s

Type	Field	Description
UINT8[4]	Resv	Reserved

Table 5-84. Definition of AMP\_DMX\_PARAM\_IMAGE\_s for EXIF SDK6 API AmpExifDmx\_Create().

Confidential  
For HAOTEK Only

## 5.4.2 AmpExifDmx\_Delete

### API Syntax:

**AmpExifDmx\_Delete** (AMP\_DMx\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete an ExifDmx handler.

### Parameters:

Type	Parameter	Description
AMP_DMx_FORMAT_HDLR_s *	hdlr	The ExifDmx handler being deleted. ( <b>AMP_DMx_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.4.1.2</a> for more details.

Table 5-85. Parameters for EXIF SDK6 API **AmpExifDmx\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-86. Returns for EXIF SDK6 API **AmpExifDmx\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.4.3 AmpExifDmx\_GetDefaultCfg

#### API Syntax:

**AmpExifDmx\_GetDefaultCfg** (AMP\_EXIF\_DMX\_CFG\_s \* config)

#### Function Description:

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

#### Parameters:

Type	Parameter	Description
AMP_EXIF_DMX_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_EXIF_DMX_CFG_s</b> is defined in <code>ExifDmx.h</code> ) Please refer to <a href="#">Section 5.4.1.1</a> for more details.

Table 5-87. Parameters for EXIF SDK6 API **AmpExifDmx\_GetDefaultCfg()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-88. Returns for EXIF SDK6 API **AmpExifDmx\_GetDefaultCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.4.4 AmpExifDmx\_GetInitDefaultCfg

### API Syntax:

**AmpExifDmx\_GetInitDefaultCfg** (AMP\_EXIF\_DMX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the ExifDmx module.

### Parameters:

Type	Parameter	Description
AMP_EXIF_DMX_INIT_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_EXIF_DMX_INIT_CFG_s</b> is defined in <code>ExifDmx.h</code> ) Please refer to <a href="#">Section 5.4.4.1</a> below for more details.

Table 5-89. Parameters for EXIF SDK6 API **AmpExifDmx\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-90. Returns for EXIF SDK6 API **AmpExifDmx\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.4.4.1 AmpExifDMX\_GetInitDefaultCfg > AMP\_EXIF\_DMX\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the ExifDmx module
UINT32	<b>BufferSize</b>	The size of the work buffer
UINT8	<b>MaxHdlr</b>	The maximum number of ExifDmx handlers

Table 5-91. Definition **AMP\_EXIF\_DMX\_INIT\_CFG\_s** of for EXIF SDK6 API **AmpExifDmx\_GetInitDefaultCfg()**.

## 5.4.5 AmpExifDmx\_GetRequiredBufferSize

### API Syntax:

**AmpExifDmx\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the ExifDmx module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of ExifDmx handlers

Table 5-92. Parameters for EXIF SDK6 API **AmpExifDmx\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 5-93. Returns for EXIF SDK6 API **AmpExifDmx\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

None



## 5.4.6 AmpExifDmx\_Init

### API Syntax:

**AmpExifDmx\_Init** (AMP\_EXIF\_DMx\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initiate the ExifDmx module.

### Parameters:

Type	Parameter	Description
AMP_EXIF_DMx_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the module. ( <b>AMP_EXIF_DMx_INIT_CFG_s</b> is defined in <code>ExifDmx.h</code> ) Please refer to <a href="#">Section 5.4.4.1</a> for more details.

Table 5-94. Parameters for EXIF SDK6 API **AmpExifDmx\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-95. Returns for EXIF SDK6 API **AmpExifDmx\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.4.7 AmpExifDmx\_Parse

### API Syntax:

**AmpExifDmx\_Parse** (AMP\_MEDIA\_INFO\_s \* media, AMP\_STREAM\_HDLR\_s \* stream)

### Function Description:

- This function is used to parse a media data from an I/O stream, and pack the data into a media information object. (Please refer to **AMP\_DMx\_FORMAT\_PARSE\_FP**).

### Parameters:

Type	Parameter	Description
AMP_MEDIA_INFO_s *	<b>media</b>	The returned media information object. ( <b>AMP_MEDIA_INFO_s</b> is defined in <code>Format.h</code> ) (Please refer to <a href="#">Section 5.4.1.4</a> )
AMP_STREAM_HDLR_s *	<b>stream</b>	The I/O stream. ( <b>AMP_STREAM_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 8.2.1.2</a> for more details.

Table 5-96. Parameters for EXIF SDK6 API **AmpExifDmx\_Parse()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-97. Returns for EXIF SDK6 API **AmpExifDmx\_Parse()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.4.8 AmpExifMux\_Create

### API Syntax:

**AmpExifMux\_Create** (AMP\_EXIF\_MUX\_CFG\_s \* config, AMP\_MUX\_FORMAT\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create an ExifMux handler.

### Parameters:

Type	Parameter	Description
AMP_EXIF_MUX_CFG_s *	<b>config</b>	The configuration used to create an ExifMux handler. Please refer to <a href="#">Section 5.4.8.1</a> below for more details.
AMP_MUX_FORMAT_HDLR_s **	<b>hdlr</b>	The returned ExifMux handler. ( <b>AMP_MUX_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.3.2.1</a> for more details.

Table 5-98. Parameters for EXIF SDK6 API **AmpExifMux\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-99. Returns for EXIF SDK6 API **AmpExifMux\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.4.8.1 AmpExifMux\_Create > AMP\_EXIF\_MUX\_CFG\_s

Type	Field	Description
AMP_STREAM_HDLR_s *	<b>Stream</b>	Stream handler (Please refer to <a href="#">Section 8.2.1.2</a> )
AMP_CFG_TAG_INFO_s	<b>SetTagInfo</b>	Tag information (Please refer to <a href="#">Section 5.4.8.2</a> )
UINT8	<b>Endian</b>	The value indicating that the Exif header is the big endian or the little endian: 0x00: Big endian 0x01: Little endian

Table 5-100. Definition of **AMP\_EXIF\_MUX\_CFG\_s** for EXIF SDK6 API **AmpExifMux\_Create()**.

#### 5.4.8.2 AmpExifMux\_Create > AMP\_CFG\_TAG\_INFO\_s

Type	Field	Description
UINT16	<b>Ifd0Tags</b>	The number of Ifd0 tags
UINT16	<b>ExifIfdTags</b>	The number of ExifIfd tags
UINT16	<b>IntIfdTags</b>	The number of IntIfd tags
UINT16	<b>Ifd1Tags</b>	The number of Ifd1 tags
UINT16	<b>GpsIfdTags</b>	The number of GPSIfd tags
AMP_CFG_TAG_s[IFD0_TOTAL_TAGS]	<b>Ifd0</b>	Ifd0 tags (Please refer to <a href="#">Section 5.4.8.3</a> )
AMP_CFG_TAG_s[EXIF_TOTAL_TAGS]	<b>ExifIfd</b>	ExtIfd tags (Please refer to <a href="#">Section 5.4.8.3</a> )
AMP_CFG_TAG_s[IntIfd_TOTAL_TAGS]	<b>IntIfd</b>	IntIfd tags (Please refer to <a href="#">Section 5.4.8.3</a> )
AMP_CFG_TAG_s[IFD1_TOTAL_TAGS]	<b>Ifd1</b>	Ifd1 tags (Please refer to <a href="#">Section 5.4.8.3</a> )
AMP_CFG_TAG_s[GPS_TOTAL_TAGS]	<b>GpsIfd</b>	GPSIfd tags (Please refer to <a href="#">Section 5.4.8.3</a> )

Table 5-101. Definition of **AMP\_CFG\_TAG\_INFO\_s** for EXIF SDK6 API **AmpExifMux\_Create()**.

#### 5.4.8.3 AmpExifMux\_Create > AMP\_CFG\_TAG\_s

Type	Field	Description
UINT8*	<b>Data</b>	Data
UINT32	<b>Value</b>	Value
UINT32	<b>Count</b>	Count
UINT16	<b>Tag</b>	Tag
UINT16	<b>Type</b>	Tag type
UINT8	<b>Set</b>	Set

Table 5-102. Definition of **AMP\_CFG\_TAG\_s** for EXIF SDK6 API **AmpExifMux\_Create()**.

## 5.4.9 AmpExifMux\_Delete

### API Syntax:

**AmpExifMux\_Delete** (AMP\_MUX\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete an ExifMux handler.

### Parameters:

Type	Parameter	Description
AMP_MUX_FORMAT_HDLR_s *	hdlr	The ExifMux handler being deleted. ( <b>AMP_MUX_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.3.2.1</a> for definitoin.

Table 5-103. Parameters for EXIF SDK6 API **AmpExifMux\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-104. Returns for EXIF SDK6 API **AmpExifMux\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.4.10 AmpExifMux\_GetDefaultCfg

### API Syntax:

**AmpExifMux\_GetDefaultCfg** (AMP\_EXIF\_MUX\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration of an ExifMux handler.

### Parameters:

Type	Parameter	Description
AMP_EXIF_MUX_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_EXIF_MUX_CFG_s</b> is defined in <code>ExifMux.h</code> ) Please refer to <a href="#">Section 5.4.8.1</a> for more details.

Table 5-105. Parameters for EXIF SDK6 API **AmpExifMux\_GetDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-106. Returns for EXIF SDK6 API **AmpExifMux\_GetDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.4.11 AmpExifMux\_GetInitDefaultCfg

#### API Syntax:

**AmpExifMux\_GetInitDefaultCfg** (AMP\_EXIF\_MUX\_INIT\_CFG\_s \* config)

#### Function Description:

- This function is used get the default configuration for initializing ExifMux module.

#### Parameters:

Type	Parameter	Description
AMP_EXIF_MUX_INIT_CFG_s *	<b>config</b>	The returned configuration ( <b>AMP_EXIF_MUX_INIT_CFG_s</b> is defined in <code>ExifMux.h</code> ) Please refer to <a href="#">Section 5.4.11.1</a> for more details.

Table 5-107. Parameters for EXIF SDK6 API **AmpExifMux\_GetInitDefaultCfg()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-108. Returns for EXIF SDK6 API **AmpExifMux\_GetInitDefaultCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

#### 5.4.11.1 AmpExifMux\_GetInitDefaultCfg > AMP\_EXIF\_MUX\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the ExifMux module
UINT32	<b>BufferSize</b>	The size of the work buffer
UINT32	<b>ExifHeadSize</b>	The header size of the module
UINT8	<b>MaxHdlr</b>	The maximum number of ExifMux handlers

Table 5-109. Definition of **AMP\_EXIF\_MUX\_INIT\_CFG\_s** for EXIF SDK6 API **AmpExifMux\_GetInitDefaultCfg()**.

## 5.4.12 AmpExifMux\_GetRequiredBufferSize

### API Syntax:

**AmpExifMux\_GetRequiredBufferSize** (UINT8 maxHdlr, UINT32 headSize)

### Function Description:

- This function is used get the required buffer size for initializing the ExifMux module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of ExifMux handlers
UINT32	<b>headSize</b>	The size of Exif header

Table 5-110. Parameters for EXIF SDK6 API **AmpExifMux\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 5-111. Returns for EXIF SDK6 API **AmpExifMux\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

None



### 5.4.13 AmpExifMux\_Init

#### API Syntax:

**AmpExifMux\_Init** (AMP\_EXIF\_MUX\_INIT\_CFG\_s \* config)

#### Function Description:

- This function is used to initialize the ExifMux module.

#### Parameters:

Type	Parameter	Description
AMP_EXIF_MUX_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the module. ( <b>AMP_EXIF_MUX_INIT_CFG_s</b> is defined in <code>ExifMux.h</code> ) Please refer to <a href="#">Section 5.4.11.1</a> for more details.

Table 5-112. Parameters for EXIF SDK6 API **AmpExifMux\_Init()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-113. Returns for EXIF SDK6 API **AmpExifMux\_Init()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.5 EXT

This section introduces the implementation of the EXT Muxing/Demuxing Format module.

The ExtMux/ExtDmx module includes the following functions:

1. Initiate the ExtMux/ExtDmx module
2. Create the ExtMux/ExtDmx handler
3. Delete the ExtMux/ExtDmx handler

Confidential  
For HAOTEK Only

## 5.5.1 AmpExtDmx\_Create

### API Syntax:

**AmpExtDmx\_Create** (AMP\_EXT\_DMx\_CFG\_s \* config, AMP\_DMx\_FORMAT\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create an ExtDmx handler.

### Parameters:

Type	Parameter	Description
AMP_EXT_DMx_CFG_s *	<b>config</b>	The configuration used to create an ExtDmx handler. ( <b>AMP_EXIF_MUX_CFG_s</b> is defined in <code>ExifMux.h</code> ) Please refer to <a href="#">Section 5.5.3.1</a> for more details.
AMP_DMx_FORMAT_HDLR_s **	<b>hdlr</b>	The returned ExtDmx handler. ( <b>AMP_DMx_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.4.1.2</a> for more details.

Table 5-114. Parameters for EXT SDK6 API **AmpExtDmx\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-115. Returns for EXT SDK6 API **AmpExtDmx\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.5.2 AmpExtDmx\_Delete

### API Syntax:

**AmpExtDmx\_Delete** (AMP\_DMX\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete an ExtDmx handler.

### Parameters:

Type	Parameter	Description
AMP_DMX_FORMAT_HDLR_s *	hdlr	The ExtDmx handler being deleted. ( <b>AMP_DMX_FORMAT_HDLR_s</b> is defined in <code>ExifMux.h</code> ) Please refer to <a href="#">Section 5.4.1.2</a> for more details.

Table 5-116. Parameters for EXT SDK6 API **AmpExtDmx\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-117. Returns for EXT SDK6 API **AmpExtDmx\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.5.3 AmpExtDmx\_GetDefaultCfg

#### API Syntax:

**AmpExtDmx\_GetDefaultCfg** (UINT8 mediaType, AMP\_EXT\_DMX\_CFG\_s \* config)

#### Function Description:

- This function is used to get the default configuration of an ExtDmx handler.

#### Parameters:

Type	Parameter	Description
UINT8	<b>mediaType</b>	The media type (Please refer to AMP_MEDIA_INFO_TYPE_e)
AMP_EXT_DMX_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_EXT_DMX_CFG_s</b> is defined in <code>ExtMux.h</code> ) Please refer to <a href="#">Section 5.5.3.1</a> for more details.

Table 5-118. Parameters for EXT SDK6 API **AmpExtDmx\_GetDefaultCfg()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-119. Returns for EXT SDK6 API **AmpExtDmx\_GetDefaultCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

#### 5.5.3.1 AmpExtDmx\_GetDefaultCfg > AMP\_EXT\_DMX\_CFG\_s

Type	Field	Description
AMP_STREAM_HDLR_s *	<b>Stream</b>	Stream handler. (The stream handler of a file being demuxed. Please refer to Section 8.2.1.2)
UINT8	<b>MediaType</b>	Media type. (The media type of a file being demuxed, for example, movie and sound)

Table 5-120. Definition of **AMP\_EXT\_DMX\_CFG\_s** for EXT SDK6 API **AmpExtDmx\_GetDefaultCfg()**.

## 5.5.4 AmpExtDmx\_GetInitDefaultCfg

### API Syntax:

**AmpExtDmx\_GetInitDefaultCfg** (AMP\_EXT\_DMX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the ExtDmx module.

### Parameters:

Type	Parameter	Description
AMP_EXT_DMX_INIT_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_EXT_DMX_INIT_CFG_s</b> is defined in <code>ExtMux.h</code> ) Please refer to <a href="#">Section 5.5.4.1</a> for more details.

Table 5-121. Parameters for EXT SDK6 API **AmpExtDmx\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-122. Returns for EXT SDK6 API **AmpExtDmx\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.5.4.1 AmpExrDmx\_GetInitDefaultCfg > AMP\_EXT\_DMX\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the ExtDmx module
UINT32	<b>BufferSize</b>	The work buffer size of the module
UINT8	<b>MaxHdlr</b>	The maximum number of ExtDmx handlers

Table 5-123. Definition of **AMP\_EXT\_DMX\_INIT\_CFG\_s** for EXT SDK6 API **AmpExtDmx\_GetInitDefaultCfg()**.

## 5.5.5 AmpExtDmx\_GetRequiredBufferSize

### API Syntax:

**AmpExtDmx\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the ExtDmx module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of ExtDmx handlers

Table 5-124. Parameters for EXT SDK6 API **AmpExtDmx\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 5-125. Returns for EXT SDK6 API **AmpExtDmx\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 5.5.6 AmpExtDmx\_Init

### API Syntax:

**AmpExtDmx\_Init** (AMP\_EXT\_DMUX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initialize the ExtDmx module.

### Parameters:

Type	Parameter	Description
AMP_EXT_DMUX_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the module. ( <b>AMP_EXT_DMUX_INIT_CFG_s</b> is defined in <code>ExtMux.h</code> ) Please refer to <a href="#">Section 5.5.4.1</a> for more details.

Table 5-126. Parameters for EXT SDK6 API **AmpExtDmx\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-127. Returns for EXT SDK6 API **AmpExtDmx\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 5.5.7 AmpExtDmx\_Parse

### API Syntax:

**AmpExtDmx\_Parse** (AMP\_MEDIA\_INFO\_s \* media, AMP\_STREAM\_HDLR\_s \* stream)

### Function Description:

- This function is used to parse media data from a stream, and pack the data into a Media information object (Please refer to AMP\_DMx\_FORMAT\_PARSE\_FP).

### Parameters:

Type	Parameter	Description
AMP_MEDIA_INFO_s *	<b>media</b>	The returned media information object. ( <b>AMP_MEDIA_INFO_s</b> is defined in <code>Format.h</code> ) (Please refer to <a href="#">Section 5.4.1.4</a> )
AMP_STREAM_HDLR_s *	<b>stream</b>	The I/O stream. ( <b>AMP_STREAM_HDLR_s</b> is defined in <code>Stream.h</code> ). Please refer to <a href="#">Section 8.2.1.2</a> for more details.

Table 5-128. Parameters for EXT SDK6 API **AmpExtDmx\_Parse()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-129. Returns for EXT SDK6 API **AmpExtDmx\_Parse()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.5.8 AmpExtMux\_Create

### API Syntax:

**AmpExtMux\_Create** (AMP\_EXT\_MUX\_CFG\_s \* config, AMP\_MUX\_FORMAT\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create an ExtMux handler.

### Parameters:

Type	Parameter	Description
AMP_EXT_MUX_CFG_s *	<b>config</b>	The configuration used to create an ExtMux handler. ( <b>AMP_EXT_MUX_CFG_s</b> is defined in <code>ExifMux.h</code> ) Please refer to <a href="#">Section 5.5.8.1</a> below for definition.
AMP_MUX_FORMAT_HDLR_s **	<b>hdlr</b>	The returned ExtMux handler. Please refer to <a href="#">Section 5.3.2.1</a> for more details.

Table 5-130. Parameters for EXT SDK6 API **AmpExtMux\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-131. Returns for EXT SDK6 API **AmpExtMux\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.5.8.1 AmpExtMux\_Create > AMP\_EXT\_MUX\_CFG\_s

Type	Field	Description
AMP_STREAM_HDLR_s *	<b>Stream</b>	Stream handler. Please refer to <a href="#">Section 8.2.1.2</a> for more details.
Union {AMP_MUX_PARAM_MOVIE_s Movie; AMP_MUX_PARAM_SOUND_s Sound; AMP_MUX_PARAM_IMAGE_s Image}	<b>Param</b>	Movie: The muxing parameters of a movie (Please refer to <a href="#">Section 5.3.1.2</a> ) Sound: The muxing parameters of a sound (Please refer to <a href="#">Section 5.3.1.3</a> ) Image: The muxing parameters of an image (Please refer to <a href="#">Section 5.3.1.4</a> )
UINT8	<b>MediaType</b>	Media type (the media type of a file being muxed, for example, movie and sound)

Table 5-132. Definition of **AMP\_EXT\_MUX\_CFG\_s** for EXT SDK6 API **AmpExtMux\_Create()**.

## 5.5.9 AmpExtMux\_Delete

### API Syntax:

**AmpExtMux\_Delete** (AMP\_MUX\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used delete an ExtMux handler.

### Parameters:

Type	Parameter	Description
AMP_MUX_FORMAT_HDLR_s *	hdlr	The ExtMux handler being deleted. ( <b>AMP_MUX_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.3.2.1</a> for more details.

Table 5-133. Parameters for EXT SDK6 API **AmpExtMux\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-134. Returns for EXT SDK6 API **AmpExtMux\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.5.10 AmpExtMux\_GetDefaultCfg

### API Syntax:

**AmpExtMux\_GetDefaultCfg** (UINT8 mediaType, AMP\_EXT\_MUX\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration of an ExtMux handler.

### Parameters:

Type	Parameter	Description
UINT8	<b>mediaType</b>	The media type (Please refer to AMP_MEDIA_INFO_TYPE_e)
AMP_EXT_MUX_CFG_s *	<b>config</b>	The returned configuration. (AMP_EXT_MUX_CFG_s is defined in ExtMux.h) Please refer to <a href="#">Section 5.5.8.1</a> for more details.

Table 5-135. Parameters for EXT SDK6 API **AmpExtMux\_GetDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-136. Returns for EXT SDK6 API **AmpExtMux\_GetDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.5.11 AmpExtMux\_GetInitDefaultCfg

#### API Syntax:

**AmpExtMux\_GetInitDefaultCfg** (AMP\_EXT\_MUX\_INIT\_CFG\_s \* config)

#### Function Description:

- This function is used to get the default configuration for initializing the ExtMux module.

#### Parameters:

Type	Parameter	Description
AMP_EXT_MUX_INIT_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_EXT_MUX_INIT_CFG_s</b> is defined in <code>ExtMux.h</code> ) Please refer to <a href="#">Section 5.5.11.1</a> below for more details.

Table 5-137. Parameters for EXT SDK6 API **AmpExtMux\_GetInitDefaultCfg()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-138. Returns for EXT SDK6 API **AmpExtMux\_GetInitDefaultCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

#### 5.5.11.1 AmpExtMux\_GetInitDefaultCfg > AMP\_EXT\_MUX\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the ExtMux module
UINT32	<b>BufferSize</b>	The work buffer size of the module
UINT8	<b>MaxHdlr</b>	The maximum number of ExtMux handlers

Table 5-139. Parameters **AMP\_EXT\_MUX\_INIT\_CFG\_s** for EXT SDK6 API **AmpExtMux\_GetInitDefaultCfg()**.

## 5.5.12 AmpExtMux\_GetRequiredBufferSize

### API Syntax:

**AmpExtMux\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the ExtMux module.

### Parameters:

Type	Parameter	Description
UINT8	maxHdlr	The maximum number of ExtMux handlers

Table 5-140. Parameters for EXT SDK6 API **AmpExtMux\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 5-141. Returns for EXT SDK6 API **AmpExtMux\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

### 5.5.13 AmpExtMux\_Init

#### API Syntax:

**AmpExtMux\_Init** (AMP\_EXT\_MUX\_INIT\_CFG\_s \* config)

#### Function Description:

- This function is used to initialize the ExtMux module.

#### Parameters:

Type	Parameter	Description
AMP_EXT_MUX_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the module. ( <b>AMP_EXT_MUX_INIT_CFG_s</b> is defined in <code>ExtMux.h</code> ) Please refer to <a href="#">Section 5.5.11.1</a> for more details.

Table 5-142. Parameters for EXT SDK6 API **AmpExtMux\_Init()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-143. Returns for EXT SDK6 API **AmpExtMux\_Init()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 5.6 Matroska

This section describes the implementation of Matroska Muxing/Demuxing/Editing Format module. Users can use MkvMux to add data into a movie, use MkvDmx to parse data from a movie, and use MkvEdt to edit a movie.

The MkvMux/MkvDmx/MkvEdt module includes the following functions:

1. Initiate the MkvMux/MkvDmx/MkvEdt module
2. Create MkvMux/MkvDmx/MkvEdt handlers
3. Delete MkvMux/MkvDmx/MkvEdt handlers

Confidential  
For HAOTEK Only

## 5.6.1 AmpMkvDmx\_Create

### API Syntax:

**AmpMkvDmx\_Create** (AMP\_MKV\_DMx\_CFG\_s \* config, AMP\_DMx\_FORMAT\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create an MkvDmx handler.

### Parameters:

Type	Parameter	Description
AMP_MKV_DMx_CFG_s *	<b>config</b>	The configuration used to create an MkvDmx handler. ( <b>AMP_MKV_DMx_CFG_s</b> is defined in <code>MkvDmx.h</code> ) Please refer to <a href="#">Section 5.6.1.1</a> for more details.
AMP_DMx_FORMAT_HDLR_s **	<b>hdlr</b>	The returned MkvDmx handler. Please refer to <a href="#">Section 5.4.1.2</a> for more details.

Table 5-144. Parameters for Matroska SDK6 API **AmpMkvDmx\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-145. Returns for Matroska SDK6 API **AmpMkvDmx\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.6.1.1 AmpMkvDmx\_Create > AMP\_MKV\_DMx\_CFG\_s

Type	Field	Description
AMP_STREAM_HDLR_s *	<b>Stream</b>	Stream handler. Please refer to <a href="#">Section 8.2.1.2</a> for more details.

Table 5-146. Definition of **AMP\_MKV\_DMx\_CFG\_s** for Matroska SDK6 API **AmpMkvDmx\_Create()**.

## 5.6.2 AmpMkvDmx\_Delete

### API Syntax:

**AmpMkvDmx\_Delete** (AMP\_DMX\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete an MkvDmx handler.

### Parameters:

Type	Parameter	Description
AMP_DMX_FORMAT_HDLR_s *	hdlr	The MkvDmx handler being deleted. ( <b>AMP_DMX_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.4.1.2</a> for more details.

Table 5-147. Parameters for Matroska SDK6 API **AmpMkvDmx\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-148. Returns for Matroska SDK6 API **AmpMkvDmx\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.6.3 AmpMkvDmx\_GetDefaultCfg

#### API Syntax:

**AmpMkvMux\_GetDefaultCfg** (AMP\_MKV\_DMx\_CFG\_s \* config)

#### Function Description:

- This function is used to get the default configuration of an MkvDmx handler.

#### Parameters:

Type	Parameter	Description
AMP_MKV_DMx_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MKV_DMx_CFG_s</b> is defined in <code>MkvDmx.h</code> ) Please refer to <a href="#">Section 5.6.1.1</a> for more details.

Table 5-149. Parameters for Matroska SDK6 API **AmpMkvDmx\_GetDefaultCfg()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-150. Returns for Matroska SDK6 API **AmpMkvDmx\_GetDefaultCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.6.4 AmpMkvMux\_GetInitDefaultCfg

### API Syntax:

**AmpMkvDmx\_GetInitDefaultCfg** (AMP\_MKV\_DMx\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the MkvDmx module.

### Parameters:

Type	Parameter	Description
AMP_MKV_DMx_INIT_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MKV_DMx_INIT_CFG_s</b> is defined in <code>MkvDmx.h</code> ) Please refer to <a href="#">Section 5.6.4.1</a> for more details.

Table 5-151. Parameters for Matroska SDK6 API **AmpMkvDmx\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-152. Returns for Matroska SDK6 API **AmpMkvDmx\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.6.4.1 AmpMkvDmx\_GetInitDefaultCfg > AMP\_MKV\_DMx\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the MkvDmx module
UINT32	<b>BufferSize</b>	The work buffer size of the module
UINT8	<b>MaxHdlr</b>	The maximum number of MkvDmx handlers

Table 5-153. Definition of **AMP\_MKV\_DMx\_INIT\_CFG\_s** for Matroska SDK6 API **AmpMkvDmx\_GetInitDefaultCfg()**.

## 5.6.5 AmpMkvDmx\_GetRequiredBufferSize

### API Syntax:

**AmpMkvDmx\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the MkvDmx module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of MkvDmx handlers

Table 5-154. Parameters for Matroska SDK6 API **AmpMkvDmx\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 5-155. Returns for Matroska SDK6 API **AmpMkvDmx\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 5.6.6 AmpMkvDmx\_Init

### API Syntax:

**AmpMkvDmx\_Init** (AMP\_MKV\_DMx\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initialize the MkvDmx module.

### Parameters:

Type	Parameter	Description
AMP_MKV_DMx_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the module. ( <b>AMP_MKV_DMx_INIT_CFG_s</b> is defined in <code>MkvDmx.h</code> ) Please refer to <a href="#">Section 5.6.4.1</a> for more details.

Table 5-156. Parameters for Matroska SDK6 API **AmpMkvDmx\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-157. Returns for Matroska SDK6 API **AmpMkvDmx\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.6.7 AmpMkvDmx\_Parse

### API Syntax:

**AmpMkvDmx\_Parse** (AMP\_MEDIA\_INFO\_s\* media, AMP\_STREAM\_HDLR\_s\* stream)

### Function Description:

- This function is used to parse media data from a stream and pack the data into a media information object (Please refer to AMP\_DMx\_FORMAT\_PARSE\_FP).

### Parameters:

Type	Parameter	Description
AMP_MEDIA_INFO_s *	<b>media</b>	The returned media information object. ( <b>AMP_MEDIA_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.4.1.4</a>
AMP_STREAM_HDLR_s *	<b>stream</b>	The I/O stream. ( <b>AMP_STREAM_HDLR_s</b> is defined in <code>Stream.h</code> ) Please refer to <a href="#">Section 8.2.1.2</a> for more details.

Table 5-158. Parameters for Matroska SDK6 API **AmpMkvDmx\_Parse()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-159. Returns for Matroska SDK6 API **AmpMkvDmx\_Parse()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 5.6.8 AmpMkvEdt\_Create

### API Syntax:

**AmpMkvEdt\_Create** (AMP\_MKV\_EDT\_CFG\_s \* config, AMP\_EDT\_FORMAT\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create an MkvEdt handler.

### Parameters:

Type	Parameter	Description
AMP_MKV_EDT_CFG_s *	<b>config</b>	The configuration used to create an MkvEdt handler. ( <b>AMP_MKV_EDT_CFG_s</b> is defined in <code>MkvEdt.h</code> ) Please refer to <a href="#">Section 5.6.8.1</a> for more details.
AMP_EDT_FORMAT_HDLR_s**	<b>hdlr</b>	The returned MkvEdt handler. ( <b>AMP_EDT_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 4.2.2.1</a> for more details.

Table 5-160. Parameters for Matroska SDK6 API **AmpMkvEdt\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-161. Returns for Matroska SDK6 API **AmpMkvEdt\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.6.8.1 AmpMkvEdt\_Create > AMP\_MKV\_EDT\_CFG\_s

Type	Field	Description
AMP_CALLBACK_f	<b>OnEvent</b>	The event callback returning an execution result.

Table 5-162. Definition of **AMP\_MKV\_EDT\_CFG\_s** for Matroska SDK6 API **AmpMkvEdt\_Create()**.

## 5.6.9 AmpMkvEdt\_Delete

### API Syntax:

**AmpMkvEdt\_Delete** (AMP\_EDT\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete an MkvEdt handler.

### Parameters:

Type	Parameter	Description
AMP_EDT_FORMAT_HDLR_s*	hdlr	The MkvEdt handler being deleted. (AMP_EDT_FORMAT_HDLR_s is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 4.2.2.1</a> for more details.

Table 5-163. Parameters for Matroska SDK6 API **AmpMkvEdt\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-164. Returns for Matroska SDK6 API **AmpMkvEdt\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.6.10 AmpMkvEdt\_GetDefaultCfg

### API Syntax:

**AmpMkvEdt\_GetDefaultCfg** (AMP\_MKV\_EDT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration of an MkvEdt handler.

### Parameters:

Type	Parameter	Description
AMP_MKV_EDT_CFG_s*	<b>config</b>	The returned configuration. (AMP_MKV_EDT_CFG_s is defined in <code>MkvEdt.h</code> ) Please refer to <a href="#">Section 5.6.8.1</a> for more details.

Table 5-165. Parameters for Matroska SDK6 API **AmpMkvEdt\_GetDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-166. Returns for Matroska SDK6 API **AmpMkvEdt\_GetDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.6.11 AmpMkvEdt\_GetInitDefaultCfg

### API Syntax:

**AmpMkvEdt\_GetInitDefaultCfg** (AMP\_MKV\_EDT\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the MkvEdt module.

### Parameters:

Type	Parameter	Description
AMP_MKV_EDT_INIT_CFG_s*	<b>config</b>	The returned configuration. (AMP_MKV_EDT_INIT_CFG_s is defined in <code>MkvEdt.h</code> ) Please refer to <a href="#">Section 5.6.11.1</a> for more details.

Table 5-167. Parameters for Matroska SDK6 API **AmpMkvEdt\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-168. Returns for Matroska SDK6 API **AmpMkvEdt\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.6.11.1 AmpMkvEdt\_GetInitDefaultCfg > AMP\_MKV\_EDT\_INIT\_CFG\_s

Type	Field	Description
UINT8*	<b>Buffer</b>	The work buffer of the MkvEdt module
UINT32	<b>BufferSize</b>	The work buffer size of the module
UINT8	<b>MaxHdlr</b>	The maximum number of MkvEdt handlers

Table 5-169. Definition of **AMP\_MKV\_EDT\_INIT\_CFG\_s** for Matroska SDK6 API **AmpMkvEdt\_GetInitDefaultCfg()**.

## 5.6.12 AmpMkvEdt\_GetRequiredBufferSize

### API Syntax:

**AmpMkvEdt\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the MkvEdt module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of MkvEdt handlers

Table 5-170. Parameters for Matroska SDK6 API **AmpMkvEdt\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 5-171. Returns for Matroska SDK6 API **AmpMkvEdt\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.6.13 AmpMkvEdt\_Init

#### API Syntax:

**AmpMkvEdt\_Init** (AMP\_MKV\_EDT\_INIT\_CFG\_s \* config)

#### Function Description:

- This function is used to initialize the MkvEdt module.

#### Parameters:

Type	Parameter	Description
AMP_MKV_EDT_INIT_CFG_s*	<b>config</b>	The configuration used to initialize the module. (AMP_MKV_EDT_INIT_CFG_s is defined in <code>MkvEdt.h</code> ) Please refer to <a href="#">Section 5.6.11.1</a> for more details.

Table 5-172. Parameters for Matroska SDK6 API **AmpMkvEdt\_Init()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-173. Returns for Matroska SDK6 API **AmpMkvEdt\_Init()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.6.14 AmpMkvMux\_Create

### API Syntax:

**AmpMkvMux\_Create** (AMP\_MKV\_MUX\_CFG\_s \* config, AMP\_MUX\_FORMAT\_HDLR\_s\*\* hdlr)

### Function Description:

- This function is used to create an MkvMux handler.

### Parameters:

Type	Parameter	Description
AMP_MKV_MUX_CFG_s*	<b>config</b>	The configuration used to create an MkvMux handler. (AMP_MKV_MUX_CFG_s is defined in <code>MkvMux.h</code> ) Please refer to <a href="#">Section 5.6.14.1</a> for more details.
AMP_MUX_FORMAT_HDLR_s**	<b>hdlr</b>	The returned MkvMux handler. Please refer to <a href="#">Section 5.3.2.1</a> for more details

Table 5-174. Parameters for Matroska SDK6 API **AmpMkvMux\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-175. Returns for Matroska SDK6 API **AmpMkvMux\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.6.14.1 AmpMkvMux\_Create > AMP\_MKV\_MUX\_CFG\_s

Type	Field	Description
AMP_STREAM_HDLR_s*	<b>Stream</b>	Stream handler. Please refer to <a href="#">Section 8.2.1.2</a> for more details.

Table 5-176. Definition of **AMP\_MKV\_MUX\_CFG\_s** for Matroska SDK6 API **AmpMkvMux\_Create()**.

## 5.6.15 AmpMkvMux\_Delete

### API Syntax:

**AmpMkvMux\_Delete** (AMP\_MUX\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete an MkvMux handler.

### Parameters:

Type	Parameter	Description
AMP_MUX_FORMAT_HDLR_s**	hdlr	The MkvMux handler being deleted (AMP_MUX_FORMAT_HDLR_s is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.3.2.1</a> for more details

Table 5-177. Parameters for Matroska SDK6 API **AmpMkvMux\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-178. Returns for Matroska SDK6 API **AmpMkvMux\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 5.6.16 AmpMkvMux\_Get DefaultCfg

### API Syntax:

**AmpMkvMux\_GetDefaultCfg** (AMP\_MKV\_MUX\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration of an MkvMux handler.

### Parameters:

Type	Parameter	Description
AMP_MKV_MUX_CFG_s*	<b>config</b>	The returned configuration. (AMP_MKV_MUX_CFG_s is defined in <code>MkvMux.h</code> ) Please refer to <a href="#">Section 5.6.14.1</a> for more details.

Table 5-179. Parameters for Matroska SDK6 API **AmpMkvMux\_GetDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-180. Returns for Matroska SDK6 API **AmpMkvMux\_GetDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.6.17 AmpMkvMux\_GetInitDefaultCfg

### API Syntax:

**AmpMkvMux\_GetInitDefaultCfg** (AMP\_MKV\_MUX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the MkvMux module.

### Parameters:

Type	Parameter	Description
AMP_MKV_MUX_INIT_CFG_s*	<b>config</b>	The returned configuration. (AMP_MKV_MUX_INIT_CFG_s is defined in <code>MkvMux.h</code> ) Please refer to <a href="#">Section 5.6.17.1</a> for more details.

Table 5-181. Parameters for Matroska SDK6 API **AmpMkvMux\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-182. Returns for Matroska SDK6 API **AmpMkvMux\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.6.17.1 AmpMkvMux\_GetInitDefaultCfg > AMP\_MKV\_MUX\_INIT\_CFG\_s

Type	Field	Description
UINT8*	<b>Buffer</b>	The work buffer of the MkvMux module
UINT32	<b>BufferSize</b>	The work buffer size of the module
UINT8	<b>MaxHdlr</b>	The maximum number of MkvMux handlers

Table 5-183. Definition of **AMP\_MKV\_MUX\_INIT\_CFG\_s** for Matroska SDK6 API **AmpMkvMux\_GetInitDefaultCfg()**.

## 5.6.18 AmpMkvMux\_GetRequiredBufferSize

### API Syntax:

**AmpMkvMux\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the MkvMux module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of MkvMux handlers

Table 5-184. Parameters for Matroska SDK6 API **AmpMkvMux\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 5-185. Returns for Matroska SDK6 API **AmpMkvMux\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.6.19 AmpMkvMux\_Init

### API Syntax:

**AmpMkvMux\_Init** (AMP\_MKV\_MUX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initialize the MkvMux module.

### Parameters:

Type	Parameter	Description
AMP_MKV_MUX_INIT_CFG_s*	<b>config</b>	The configuration used to initialize the module. (AMP_MKV_MUX_INIT_CFG_s is defined in <code>MkvMux.h</code> ) (Please refer to <a href="#">Section 5.6.17.1</a> for more details)

Table 5-186. Parameters for Matroska SDK6 API **AmpMkvMux\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-187. Returns for Matroska SDK6 API **AmpMkvMux\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.7 MOV

This section explains the implementation of MOV Muxing/Demuxing/Editing Format module. Users can use MovMux to add data into a movie, use MovDmx to parse data from a movie, and use MovEdt to edit a movie.

The MovMux/MovDmx/MovEdt module includes the following functions:

1. Initiate the MovMux/MovDmx/MovEdt module
2. Create MovMux/MovDmx/MovEdt handlers
3. Delete MovMux/MovDmx/MovEdt handlers

Confidential  
For HAOTEK Only

## 5.7.1 AmpMovDmx\_Create

### API Syntax:

**AmpMovDmx\_Create** (AMP\_MOV\_DMX\_CFG\_s \* config, AMP\_DMX\_FORMAT\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create an MovDmx handler.

### Parameters:

Type	Parameter	Description
AMP_MOV_DMX_CFG_s *	<b>config</b>	The configuration used to create an MovDmx handler. ( <b>AMP_MOV_DMX_CFG_s</b> is defined in <code>MovDmx.h</code> ) Please refer to <a href="#">Section 5.7.1.1</a> for more details.
AMP_DMX_FORMAT_HDLR_s **	<b>hdlr</b>	The returned MovDmx handler. Please refer to <a href="#">Section 5.4.1.2</a> for more details.

Table 5-188. Parameters for MOV SDK6 API **AmpMovDmx\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-189. Returns for MOV SDK6 API **AmpMovDmx\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.7.1.1 Amp\_MovDmx\_Create > AMP\_MOV\_DMX\_CFG\_s

Type	Field	Description
AMP_STREAM_HDLR_s *	<b>Stream</b>	Stream handler (Please refer to <a href="#">Section 8.2.1.2</a> )
AMP_MOV_DMX_CFG_INFO_s	<b>DmxCfgInfo</b>	The configuration of creating an MovDmx handler (Please refer to <a href="#">Section 5.7.1.2</a> )

Table 5-190. Definition of **AMP\_MOV\_DMX\_CFG\_s** for MOV SDK6 API **AmpMovDmx\_Create()**.

### 5.7.1.2 Amp\_MovDmx\_Create > AMP\_MOV\_DMX\_CFG\_INFO\_s

Type	Field	Description
UINT32	<b>MaxIdxNum</b>	The maximum number of indexes of a MovDmx handler

Table 5-191. Definition of **AMP\_MOV\_DMX\_CFG\_INFO\_s** for MP4 SDK6 API **AmpMovDmx\_Create()**.

Confidential  
For HAOTEK Only

## 5.7.2 AmpMovDmx\_Delete

### API Syntax:

**AmpMovDmx\_Delete** (AMP\_DMX\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete an MovDmx handler.

### Parameters:

Type	Parameter	Description
AMP_DMX_FORMAT_HDLR_s *	hdlr	The MovDmx handler being deleted. ( <b>AMP_DMX_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.4.1.2</a> for more details.

Table 5-192. Parameters for MOV SDK6 API **AmpMovDmx\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-193. Returns for MOV SDK6 API **AmpMovDmx\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



### 5.7.3 AmpMovDmx\_GetDefaultCfg

#### API Syntax:

**AmpMovDmx\_GetDefaultCfg**(AMP\_MOV\_DMX\_CFG\_s \* config)

#### Function Description:

- This function is used to get the default configuration of an MovDmx handler.

#### Parameters:

Type	Parameter	Description
AMP_MOV_DMX_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MOV_DMX_CFG_s</b> is defined in <code>MovDmx.h</code> ) Please refer to <a href="#">Section 5.7.1.1</a> for more details.

Table 5-194. Parameters for MOV SDK6 API **AmpMovDmx\_GetDefaultCfg()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-195. Returns for MOV SDK6 API **AmpMovDmx\_GetDefaultCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.7.4 AmpMovDmx\_GetInitDefaultCfg

### API Syntax:

**AmpMovDmx\_GetInitDefaultCfg** (AMP\_MOV\_DMX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the MovDmx module.

### Parameters:

Type	Parameter	Description
AMP_MOV_DMX_INIT_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MOV_DMX_INIT_CFG_s</b> is defined in <code>MovDmx.h</code> ) Please refer to <a href="#">Section 5.7.6.1</a> for more details.

Table 5-196. Parameters for MOV SDK6 API **AmpMovDmx\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-197. Returns for MOV SDK6 API **AmpMovDmx\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.7.5 AmpMovDmx\_GetRequiredBufferSize

### API Syntax:

**AmpMovDmx\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the MovDmx module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of MovDmx handlers

Table 5-198. Parameters for MOV SDK6 API **AmpMovDmx\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size.

Table 5-199. Returns for MOV SDK6 API **AmpMovDmx\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 5.7.6 AmpMovDmx\_Init

### API Syntax:

**AmpMovDmx\_Init** (AMP\_MOV\_DMX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initialize the MovDmx module.

### Parameters:

Type	Parameter	Description
AMP_MOV_DMX_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the module. ( <b>AMP_MOV_DMX_INIT_CFG_s</b> is defined in <code>MovDmx.h</code> ) Please refer to <a href="#">Section 5.7.6.1</a> below for more details.

Table 5-200. Parameters for MOV SDK6 API **AmpMovDmx\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-201. Returns for MOV SDK6 API **AmpMovDmx\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.7.6.1 AmpMovDmx\_Init > AMP\_MOV\_DMX\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the MovDmx module
UINT32	<b>BufferSize</b>	The work buffer size of the module
UINT8	<b>MaxHdlr</b>	The maximum number of MovDmx handlers

Table 5-202. Definition of **AMP\_MOV\_DMX\_INIT\_CFG\_s** for MOV SDK6 API **AmpMovDmx\_Init()**.

## 5.7.7 AmpMovDmx\_Parse

### API Syntax:

**AmpMovDmx\_Parse** (AMP\_MEDIA\_INFO\_s \* media, AMP\_STREAM\_HDLR\_s \* stream)

### Function Description:

- This function is used to parse media data from a stream, and pack the data into a Media information object. (Please refer to **AMP\_DMx\_FORMAT\_PARSE\_FP**).

### Parameters:

Type	Parameter	Description
AMP_MEDIA_INFO_s *	<b>media</b>	The returned Media information object. ( <b>AMP_MEDIA_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.4.1.4</a> for more details.
AMP_STREAM_HDLR_s *	<b>stream</b>	The I/O stream. Please refer to <a href="#">Section 8.2.1.2</a> for more details.

Table 5-203. Parameters for MOV SDK6 API **AmpMovDmx\_Parse()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-204. Returns for MOV SDK6 API **AmpMovDmx\_Parse()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.7.8 AmpMovEdt\_Create

### API Syntax:

**AmpMovEdt\_Create** (AMP\_MOV\_EDT\_CFG\_s \* config, AMP\_EDT\_FORMAT\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create an MovEdt handler.

### Parameters:

Type	Parameter	Description
AMP_MOV_EDT_CFG_s *	<b>config</b>	The configuration used to create an MovEdt handler. ( <b>AMP_MOV_EDT_CFG_s</b> is defined in <code>MovEdt.h</code> ) Please refer to <a href="#">Section 5.7.8.1</a> below for more details.
AMP_EDT_FORMAT_HDLR_s **	<b>hdlr</b>	The returned MovEdt handler. ( <b>AMP_EDT_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 4.2.2.1</a> for more details.

Table 5-205. Parameters for MOV SDK6 API **AmpMovEdt\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-206. Returns for MOV SDK6 API **AmpMovEdt\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.7.8.1 AmpMovEdt\_Create > AMP\_MOV\_EDT\_CFG\_s

Type	Field	Description
AMP_INDEX_HDLR_s *	<b>Index</b>	Index handler. Please refer to <a href="#">Section 7.2.6.2</a> .
AMP_CALLBACK_f	<b>OnEvent</b>	The event callback returning an execution result
BOOL8	<b>EnableMoovHead</b>	The flag to enable the flow of putting MOOV before MDAT

Table 5-207. Definition of **AMP\_MOV\_EDT\_CFG\_s** for MOV SDK6 API **AmpMovEdt\_Create()**.

Confidential  
For HAOTEK Only

## 5.7.9 AmpMovEdt\_Delete

### API Syntax:

**AmpMovEdt\_Delete** (AMP\_EDT\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete an MovEdt handler.

### Parameters:

Type	Parameter	Description
AMP_EDT_FORMAT_HDLR_s *	hdlr	The MovEdt handler being deleted. ( <b>AMP_EDT_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 4.2.2.1</a> for more details.

Table 5-208. Parameters for MOV SDK6 API **AmpMovEdt\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-209. Returns for MOV SDK6 API **AmpMovEdt\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 5.7.10 AmpMovEdt\_GetDefaultCfg

### API Syntax:

**AmpMovEdt\_GetDefaultCfg** (AMP\_MOV\_EDT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration of an MovEdt handler.

### Parameters:

Type	Parameter	Description
AMP_MOV_EDT_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MOV_EDT_CFG_s</b> is defined in <code>MovEdt.h</code> ) Please refer to <a href="#">Section 5.7.8.1</a> for more details.

Table 5-210. Parameters for MOV SDK6 API **AmpMovEdt\_GetDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-211. Returns for MOV SDK6 API **AmpMovEdt\_GetDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.7.11 AmpMovEdt\_GetInitDefaultCfg

#### API Syntax:

**AmpMovEdt\_GetInitDefaultCfg** (AMP\_MOV\_EDT\_INIT\_CFG\_s \* config)

#### Function Description:

- This function is used to get the default configuration for initializing the MovEdt module.

#### Parameters:

Type	Parameter	Description
AMP_MOV_EDT_INIT_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MOV_EDT_INIT_CFG_s</b> is defined in <code>MovEdt.h</code> ) Please refer to <a href="#">Section 5.7.11.1</a> below for more details.

Table 5-212. Parameters for MOV SDK6 API **AmpMovEdt\_GetInitDefaultCfg()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-213. Returns for MOV SDK6 API **AmpMovEdt\_GetInitDefaultCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

#### 5.7.11.1 AmpMovEdt\_GetInitDefaultCfg > AMP\_MOV\_EDT\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the MovMux module
UINT32	<b>BufferSize</b>	The work buffer size of the module
UINT8	<b>MaxHdlr</b>	The maximum number of MovMux handlers

Table 5-214. Definition of **AMP\_MOV\_EDT\_INIT\_CFG\_s** for MOV SDK6 API **AmpMovEdt\_GetInitDefaultCfg()**.

## 5.7.12 AmpMovEdt\_GetRequiredBufferSize

### API Syntax:

**AmpMovEdt\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the MovEdt module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of MovEdt handlers

Table 5-215. Parameters for MOV SDK6 API **AmpMovEdt\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 5-216. Returns for MOV SDK6 API **AmpMovEdt\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.7.13 AmpMovEdt\_Init

#### API Syntax:

**AmpMovEdt\_Init** (AMP\_MOV\_EDT\_INIT\_CFG\_s \* config)

#### Function Description:

- This function is used for the initialize the MovEdt module.

#### Parameters:

Type	Parameter	Description
AMP_MOV_EDT_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the module. ( <b>AMP_MOV_EDT_INIT_CFG_s</b> is defined in <code>MovEdt.h</code> ) Please refer to <a href="#">Section 5.7.11.1</a> below for more details.

Table 5-217. Parameters for MOV SDK6 API **AmpMovEdt\_Init()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-218. Returns for MOV SDK6 API **AmpMovEdt\_Init()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.7.14 AmpMovMux\_Create

### API Syntax:

**AmpMovMux\_Create** (AMP\_MOV\_MUX\_CFG\_s \* config, AMP\_MUX\_FORMAT\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create an MovMux handler.

### Parameters:

Type	Parameter	Description
AMP_MOV_MUX_CFG_s *	<b>config</b>	The configuration used to create an MovMux handler. ( <b>AMP_MOV_MUX_CFG_s</b> is defined in <code>MovEdt.h</code> ) Please refer to <a href="#">Section 5.7.14.1</a> below for more details.
AMP_MUX_FORMAT_HDLR_s **	<b>hdlr</b>	The returned MovMux handler. ( <b>AMP_MUX_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.3.2.1</a> for more details.

Table 5-219. Parameters for MOV SDK6 API **AmpMovMux\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-220. Returns for MOV SDK6 API **AmpMovMux\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

#### 5.7.14.1 AmpMovMux\_Create > AMP\_MOV\_MUX\_CFG\_s

Type	Field	Description
AMP_STREAM_HDLR_s*	<b>Stream</b>	Stream handler. Please refer to <a href="#">Section 8.2.1.2</a> .
AMP_INDEX_HDLR_s *	<b>Index</b>	Index handler. Please refer to <a href="#">Section 7.2.6.2</a> .
UINT32	<b>MaxIdxNum</b>	The maximum number of the cached index of the MovMux handler
UINT32	<b>TrickRecDivisor</b>	The divisor factor to decide media fps for High frame rate. If fps is 120 and TrickRecDivisor is 4, then the container will be 30 fps.
AMP_ISO_PUT_MOOV_FP	<b>PutMoov</b>	The callback of MOOV putting function
AMO_ISO_GET_MOOV_SIZE_FP	<b>GetMoovSize</b>	The callback of getting MOOV size function
BOOL8	<b>EnableMoovHead</b>	The flag to enable the flow of putting MOOV before MDAT
BOOL8	<b>EnableCO64</b>	The flag to support large MDAT offset (64-bit file)

Table 5-221. Definition of AMP\_MOV\_MUX\_CFG\_s for MOV SDK6 API *AmpMovMux\_Create()*.

## 5.7.15 AmpMovMux\_Delete

### API Syntax:

**AmpMovMux\_Delete** (AMP\_MUX\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete an MovMux handler.

### Parameters:

Type	Parameter	Description
AMP_MUX_FORMAT_HDLR_s *	hdlr	The MovMux handler being deleted. ( <b>AMP_MUX_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.3.2.1</a> for more details.

Table 5-222. Parameters for MOV SDK6 API **AmpMovMux\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-223. Returns for MOV SDK6 API **AmpMovMux\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.7.16 AmpMovMux\_GetDefaultCfg

### API Syntax:

**AmpMovMux\_GetDefaultCfg** (AMP\_MOV\_MUX\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration of an MovMux handler.

### Parameters:

Type	Parameter	Description
AMP_MOV_MUX_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MOV_MUX_CFG_s</b> is defined in <code>MovMux.h</code> ) Please refer to <a href="#">Section 5.7.14.1</a> for more details.

Table 5-224. Parameters for MOV SDK6 API **AmpMovMux\_GetDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-225. Returns for MOV SDK6 API **AmpMovMux\_GetDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 5.7.17 AmpMovMux\_GetInitDefaultCfg

### API Syntax:

**AmpMovMux\_GetInitDefaultCfg** (AMP\_MOV\_MUX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the MovMux module.

### Parameters:

Type	Parameter	Description
AMP_MOV_MUX_INIT_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MOV_MUX_INIT_CFG_s</b> is defined in <code>MovMux.h</code> ) Please refer to <a href="#">Section 5.7.17.1</a> for more details.

Table 5-226. Parameters for MOV SDK6 API **AmpMovMux\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-227. Returns for MOV SDK6 API **AmpMovMux\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.7.17.1 AmpMovMux\_GetInitDefaultCfg > AMP\_MOV\_MUX\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the MovMux module
UINT32	<b>BufferSize</b>	The work buffer size of the module
UINT8	<b>MaxHdlr</b>	The maximum number of MovMux handlers

Table 5-228. Definition of **AMP\_MOV\_MUX\_INIT\_CFG\_s** for MOV SDK6 API **AmpMovMux\_GetInitDefaultCfg()**.

## 5.7.18 AmpMovMux\_GetRequiredBufferSize

### API Syntax:

**AmpMovMux\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the MovMux module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of MovMux handlers

Table 5-229. Parameters for MOV SDK6 API **AmpMovMux\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 5-230. Returns for MOV SDK6 API **AmpMovMux\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.7.19 AmpMovMux\_Init

### API Syntax:

**AmpMovMux\_Init** (AMP\_MOV\_MUX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used for the initialize the MovMux module.

### Parameters:

Type	Parameter	Description
AMP_MOV_MUX_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the module. ( <b>AMP_MOV_MUX_INIT_CFG_s</b> is defined in <code>MovMux.h</code> ) Please refer to <a href="#">Section 5.7.17.1</a> below for more details.

Table 5-231. Parameters for MOV SDK6 API **AmpMovMux\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-232. Returns for MOV SDK6 API **AmpMovMux\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.8 MP4

This section explains the implementation of MP4 Muxing/Demuxing/Editing Format module. Users can use Mp4Mux to add data into a movie, use Mp4Dmx to parse data from a movie, and use Mp4Edt to edit a movie.

The Mp4Mux/Mp4Dmx/Mp4Edt module includes the following functions:

1. Initiate the Mp4Mux/Mp4Dmx/Mp4Edt module
2. Create Mp4Mux/Mp4Dmx/Mp4Edt handlers
3. Delete Mp4Mux/Mp4Dmx/Mp4Edt handlers

Confidential  
For HAOTEK Only

## 5.8.1 AmpMp4Dmx\_Create

### API Syntax:

**AmpMp4Dmx\_Create** (AMP\_MP4\_DMX\_CFG\_s \* config, AMP\_DMX\_FORMAT\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create an Mp4Dmx handler.

### Parameters:

Type	Parameter	Description
AMP_MP4_DMX_CFG_s *	<b>config</b>	The configuration used to create an Mp4Dmx handler. ( <b>AMP_MP4_DMX_CFG_s</b> is defined in <code>MP4Dmx.h</code> ) Please refer to <a href="#">Section 5.8.1.1</a> below for more details.
AMP_DMX_FORMAT_HDLR_s **	<b>hdlr</b>	The returned Mp4Dmx handler. ( <b>AMP_DMX_FORMAT_HDLR_s</b> is defined in <code>MP4Dmx.h</code> ) Please refer to <a href="#">Section 5.4.1.2</a> for more details.

Table 5-233. Parameters for MP4 SDK6 API **AmpMp4Dmx\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-234. Returns for MP4 SDK6 API **AmpMp4Dmx\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

### 5.8.1.1 Amp\_Mp4Dmx\_Create > AMP\_MP4\_DMX\_CFG\_s

Type	Field	Description
AMP_STREAM_HDLR_s *	<b>Stream</b>	Stream handler. Please refer to <a href="#">Section 8.2.1.2</a> .
AMP_MP4_DMX_CFG_INFO_s	<b>DmxCfgInfo</b>	The configuration of creating an Mp4Dmx handler. Please refer to <a href="#">Section 5.8.1.2</a>

Table 5-235. Definition of **AMP\_MP4\_DMX\_CFG\_s** for MP4 SDK6 API **AmpMp4Dmx\_Create()**.

### 5.8.1.2 Amp\_Mp4Dmx\_Create > AMP\_MP4\_DMx\_CFG\_INFO\_s

Type	Field	Description
UINT32	<b>MaxIdxNum</b>	The maximum number of the indexes of an Mp4Dmx handler.

Table 5-236. Definition of **AMP\_MP4\_DMx\_CFG\_INFO\_s** for MP4 SDK6 API **AmpMp4Dmx\_Create()**.

Confidential  
For HAOTEK Only

## 5.8.2 AmpMp4Dmx\_Delete

### API Syntax:

**AmpMp4Dmx\_Delete** (AMP\_DMX\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete an Mp4Dmx handler.

### Parameters:

Type	Parameter	Description
AMP_DMX_FORMAT_HDLR_s *	hdlr	The Mp4Dmx handler being deleted. ( <b>AMP_DMX_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.4.1.2</a> for more details.

Table 5-237. Parameters for MP4 SDK6 API **AmpMp4Dmx\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-238. Returns for MP4 SDK6 API **AmpMp4Dmx\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

### 5.8.3 AmpMp4Dmx\_GetDefaultCfg

#### API Syntax:

**AmpMp4Dmx\_GetDefaultCfg** (AMP\_MP4\_DMX\_CFG\_s \* config)

#### Function Description:

- This function is used to get the default configuration of an Mp4Dmx handler.

#### Parameters:

Type	Parameter	Description
AMP_MP4_DMX_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MP4_DMX_CFG_s</b> is defined in <code>Mp4Dmx.h</code> ) Please refer to <a href="#">Section 5.8.1.1</a> for more details.

Table 5-239. Parameters for MP4 SDK6 API **AmpMp4Dmx\_GetDefaultCfg()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-240. Returns for MP4 SDK6 API **AmpMp4Dmx\_GetDefaultCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

None



## 5.8.4 AmpMp4Dmx\_GetInitDefaultCfg

### API Syntax:

**AmpMp4Dmx\_GetInitDefaultCfg** (AMP\_MP4\_DMX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the Mp4Dmx module.

### Parameters:

Type	Parameter	Description
AMP_MP4_DMX_INIT_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MP4_DMX_INIT_CFG_s</b> is defined in <code>Mp4Dmx.h</code> ) Please refer to <a href="#">Section 5.8.4.1</a> below for more details.

Table 5-241. Parameters for MP4 SDK6 API **AmpMp4Dmx\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-242. Returns for MP4 SDK6 API **AmpMp4Dmx\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.8.4.1 AmpMp4Dmx\_GetInitDefaultCfg > AMP\_MP4\_DMX\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the Mp4Dmx module
UINT32	<b>BufferSize</b>	The work buffer size of the module
UINT8	<b>MaxHdlr</b>	The maximum number of Mp4Dmx handlers

Table 5-243. Definition of **AMP\_MP4\_DMX\_INIT\_CFG\_s** for MP4 SDK6 API **AmpMp4Dmx\_GetInitDefaultCfg()**.

## 5.8.5 AmpMp4Dmx\_GetRequiredBufferSize

### API Syntax:

**AmpMp4Dmx\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the Mp4Dmx module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of Mp4Dmx handlers

Table 5-244. Parameters for MP4 SDK6 API **AmpMp4Dmx\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 5-245. Returns for MP4 SDK6 API **AmpMp4Dmx\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 5.8.6 AmpMp4Dmx\_Init

### API Syntax:

**AmpMp4Dmx\_Init** (AMP\_MP4\_DMx\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initialize the Mp4Dmx module.

### Parameters:

Type	Parameter	Description
AMP_MP4_DMx_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the module. ( <b>AMP_MP4_DMx_INIT_CFG_s</b> is defined in <code>Mp4Dmx.h</code> ) Please refer to <a href="#">Section 5.8.4.1</a> for definition.

Table 5-246. Parameters for MP4 SDK6 API **AmpMp4Dmx\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-247. Returns for MP4 SDK6 API **AmpMp4Dmx\_Init()**.

### Example:

None

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.8.7 AmpMp4Dmx\_Parse

### API Syntax:

**AmpMp4Dmx\_Parse** (AMP\_MEDIA\_INFO\_s \* media, AMP\_STREAM\_HDLR\_s \* stream)

### Function Description:

- This function is used to parse media data from a stream, and pack the data into a Media information object. (Please refer to **AMP\_DMx\_FORMAT\_PARSE\_FP**).

### Parameters:

Type	Parameter	Description
AMP_MEDIA_INFO_s *	<b>media</b>	The returned Media information object. ( <b>AMP_MEDIA_INFO_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.4.1.4</a> .
AMP_STREAM_HDLR_s *	<b>stream</b>	The I/O stream. ( <b>AMP_STREAM_HDLR_s</b> is defined in <code>Stream.h</code> ) Please refer to <a href="#">Section 8.2.1.2</a> for more details.

Table 5-248. Parameters for MP4 SDK6 API **AmpMp4Dmx\_Parse()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-249. Returns for MP4 SDK6 API **AmpMp4Dmx\_Parse()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.8.8 AmpMp4Edt\_Create

### API Syntax:

**AmpMp4Edt\_Create** (AMP\_MP4\_EDT\_CFG\_s \* config, AMP\_EDT\_FORMAT\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create an Mp4Edt handler.

### Parameters:

Type	Parameter	Description
AMP_MP4_EDT_CFG_s *	<b>config</b>	The configuration used to create an Mp4Edt handler. ( <b>AMP_MP4_EDT_CFG_s</b> is defined in <code>Mp4Edt.h</code> ) Please refer to <a href="#">Section 5.8.8.1</a> below for more details.
AMP_EDT_FORMAT_HDLR_s **	<b>hdlr</b>	The returned Mp4Edt handler. ( <b>AMP_EDT_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 4.2.2.1</a> below for more details.

Table 5-250. Parameters for MP4 SDK6 API **AmpMp4Edt\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-251. Returns for MP4 SDK6 API **AmpMp4Edt\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.8.8.1 AmpMp4Edt\_Create > AMP\_MP4\_EDT\_CFG\_s

Type	Field	Description
AMP_INDEX_HDLR_s *	Index	Index handler (Please refer to <a href="#">Section 7.2.6.2</a> )
AMP_CALLBACK_f	OnEvent	The event callback returning an execution result
BOOL8	EnableMoovHead	The flag to enable the flow of putting MOOV before MDAT

Table 5-252. Definition of AMP\_MP4\_EDT\_CFG\_s for MP4 SDK6 API AmpMp4Edt\_Create().

Confidential  
For HAOTEK Only

## 5.8.9 AmpMp4Edt\_Delete

### API Syntax:

**AmpMp4Edt\_Delete** (AMP\_EDT\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete an Mp4Edt handler.

### Parameters:

Type	Parameter	Description
AMP_EDT_FORMAT_HDLR_s *	hdlr	The Mp4Edt handler being deleted. ( <b>AMP_EDT_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 4.2.2.1</a> for more details.

Table 5-253. Parameters for MP4 SDK6 API **AmpMp4Edt\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-254. Returns for MP4 SDK6 API **AmpMp4Edt\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.8.10 AmpMp4Edt\_GetDefaultCfg

### API Syntax:

**AmpMp4Edt\_GetDefaultCfg** (AMP\_MP4\_EDT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration of an Mp4Edt handler.

### Parameters:

Type	Parameter	Description
AMP_MP4_EDT_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MP4_EDT_CFG_s</b> is defined in <code>Mp4Edt.h</code> ) Please refer to <a href="#">Section 5.8.8.1</a> for more details.

Table 5-255. Parameters for MP4 SDK6 API **AmpMp4Edt\_GetDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-256. Returns for MP4 SDK6 API **AmpMp4Edt\_GetDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



### 5.8.11 AmpMp4Edt\_GetInitDefaultCfg

#### API Syntax:

**AmpMp4Edt\_GetInitDefaultCfg** (AMP\_MP4\_EDT\_INIT\_CFG\_s \* config)

#### Function Description:

- This function is used to get the default configuration for initializing the Mp4Edt module.

#### Parameters:

Type	Parameter	Description
AMP_MP4_EDT_INIT_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MP4_EDT_INIT_CFG_s</b> is defined in <code>Mp4Edt.h</code> ) Please refer to <a href="#">Section 5.8.11.1</a> for more details.

Table 5-257. Parameters for MP4 SDK6 API **AmpMp4Edt\_GetInitDefaultCfg()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-258. Returns for MP4 SDK6 API **AmpMp4Edt\_GetInitDefaultCfg()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

#### 5.8.11.1 AmpMp4Edt\_GetInitDefaultCfg > AMP\_MP4\_EDT\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the Mp4Edt module
UINT32	<b>BufferSize</b>	The work buffer size of the module
UINT8	<b>MaxHdlr</b>	The maximum number of Mp4Edt handlers

Table 5-259. Definition of **AMP\_MP4\_EDT\_INIT\_CFG\_s** for MP4 SDK6 API **AmpMp4Edt\_GetInitDefaultCfg()**.

## 5.8.12 AmpMp4Edt\_GetRequiredBufferSize

### API Syntax:

**AmpMp4Edt\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the Mp4Edt module.

### Parameters:

Type	Parameter	Description
UINT8	maxHdlr	The maximum number of Mp4Edt handlers

Table 5-260. Parameters for MP4 SDK6 API **AmpMp4Edt\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 5-261. Returns for MP4 SDK6 API **AmpMp4Edt\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

### 5.8.13 AmpMp4Edt\_Init

#### API Syntax:

**AmpMp4Edt\_Init** (AMP\_MP4\_EDT\_INIT\_CFG\_s \* config)

#### Function Description:

- This function is used to initialize the Mp4Edt module.

#### Parameters:

Type	Parameter	Description
AMP_MP4_EDT_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the module. ( <b>AMP_MP4_EDT_INIT_CFG_s</b> is defined in <code>Mp4Edt.h</code> ) Please refer to <a href="#">Section 5.8.11.1</a> for more details.

Table 5-262. Parameters for MP4 SDK6 API **AmpMp4Edt\_Init()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-263. Returns for MP4 SDK6 API **AmpMp4Edt\_Init()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.8.14 AmpMp4Mux\_Create

### API Syntax:

**AmpMp4Mux\_Create** (AMP\_MP4\_MUX\_CFG\_s \*config, AMP\_MUX\_FORMAT\_HDLR\_s \*\*hdlr)

### Function Description:

- This function is used to create an Mp4Mux handler.

### Parameters:

Type	Parameter	Description
AMP_MP4_MUX_CFG_s *	<b>config</b>	The configuration used to create an Mp4Mux handler ( <b>AMP_MP4_MUX_CFG_s</b> is defined in Mp4Mux.h) Please refer to <a href="#">Section 5.8.16.1</a> for more details.
AMP_MUX_FORMAT_HDLR_s **	<b>hdlr</b>	The returned Mp4Mux handler. ( <b>AMP_MUX_FORMAT_HDLR_s</b> is defined in Format.h) Please refer to <a href="#">Section 5.3.2.1</a> for definition.

Table 5-264. Parameters for MP4 SDK6 API **AmpMp4Mux\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-265. Returns for MP4 SDK6 API **AmpMp4Mux\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 5.8.15 AmpMp4Mux\_Delete

### API Syntax:

**AmpMp4Mux\_Delete** (AMP\_MUX\_FORMAT\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete an Mp4Mux handler.

### Parameters:

Type	Parameter	Description
AMP_MUX_FORMAT_HDLR_s *	hdlr	The Mp4Mux handler being deleted. ( <b>AMP_MUX_FORMAT_HDLR_s</b> is defined in <code>Format.h</code> ) Please refer to <a href="#">Section 5.3.2.1</a> for more details.

Table 5-266. Parameters for MP4 SDK6 API **AmpMp4Mux\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-267. Returns for MP4 SDK6 API **AmpMp4Mux\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.8.16 AmpMp4Mux\_GetDefaultCfg

**API Syntax:**

**AmpMp4Mux\_GetDefaultCfg** (AMP\_MP4\_MUX\_CFG\_s \* config)

**Function Description:**

- This function is used to get the default configuration of an Mp4Mux handler.

**Parameters:**

Type	Parameter	Description
AMP_MP4_MUX_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MP4_MUX_CFG_s</b> is defined in <code>Mp4Mux.h</code> ) Please refer to <a href="#">Section 5.8.16.1</a> below for definition.

Table 5-268. Parameters for MP4 SDK6 API **AmpMp4Mux\_GetDefaultCfg()**.

**Returns:**

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-269. Returns for MP4 SDK6 API **AmpMp4Mux\_GetDefaultCfg()**.

**Example:**

Please refer to Unit Test document.

**See Also:**

Please refer to [Chapter 10](#) for more details on error codes.

### 5.8.16.1 AmpMp4Mux\_GetDefaultCfg > AMP\_MP4\_MUX\_CFG\_s

Type	Field	Description
AMP_STREAM_HDLR_s *	<b>Stream</b>	Stream handler. Please refer to the <a href="#">Section 8.2.1.2</a> for more details.
AMP_INDEX_HDLR_s *	<b>Index</b>	Index handler. Please refer to <a href="#">Section 7.2.6.2</a> for more details.
UINT32	<b>MaxIdxNum</b>	The maximum number of the cached index of an Mp4Mux handler
UINT32	<b>TrickRecDivisor</b>	The divisor factor to decide media fps for High frame rate. If fps is 120 and TrickRecDivisor is 4, then the container will be 30 fps.
AMP_ISO_PUT_MOOV_FP	<b>PutMoov</b>	The callback of MOOV putting function
AMP_ISO_GET_MOOV_SIZE_FP	<b>GetMoovSize</b>	The callback of getting MOOV size function
BOOL8	<b>EnableMoovHead</b>	The flag to enable the flow of putting MOOV before MDAT
BOOL8	<b>EnableCO64</b>	The flag to support large MDAT offset (64-bit file)

Table 5-270. Definition of AMP\_MP4\_MUX\_CFG\_s for MP4 SDK6 API **AmpMp4Mux\_GetDefaultCfg()**.

## 5.8.17 AmpMp4Mux\_GetInitDefaultCfg

### API Syntax:

**AmpMp4Mux\_GetInitDefaultCfg** (AMP\_MP4\_MUX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the Mp4Mux module.

### Parameters:

Type	Parameter	Description
AMP_MP4_MUX_INIT_CFG_s *	<b>config</b>	The returned configuration. ( <b>AMP_MP4_MUX_INIT_CFG_s</b> is defined in <code>Mp4Mux.h</code> ) Please refer to <a href="#">Section 5.8.17.1</a> below for definition.

Table 5-271. Parameters for MP4 SDK6 API **AmpMp4Mux\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-272. Returns for MP4 SDK6 API **AmpMp4Mux\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 5.8.17.1 AmpMp4Mux\_GetInitDefaultCfg > AMP\_MP4\_MUX\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the Mp4Mux module
UINT32	<b>BufferSize</b>	The work buffer size of the module
UINT8	<b>MaxHdlr</b>	The maximum number of Mp4Mux handlers

Table 5-273. Definition of **AMP\_MP4\_MUX\_INIT\_CFG\_s** for MP4 SDK6 API **AmpMp4Mux\_GetInitDefaultCfg()**.



## 5.8.18 AmpMp4Mux\_GetRequiredBufferSize

### API Syntax:

**AmpMp4Mux\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used get the required buffer size for initializing the Mp4Mux module.

### Parameters:

Type	Parameter	Description
UINT8	maxHdlr	The maximum number of Mp4Mux handlers

Table 5-274. Parameters for MP4 SDK6 API **AmpMp4Mux\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size.

Table 5-275. Returns for MP4 SDK6 API **AmpMp4Mux\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 5.8.19 AmpMp4Mux\_Init

### API Syntax:

**AmpMp4Mux\_Init** (AMP\_MP4\_MUX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initialize the Mp4Mux module.

### Parameters:

Type	Parameter	Description
AMP_MP4_MUX_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the module. ( <b>AMP_MP4_MUX_INIT_CFG_s</b> is defined in <code>Mp4Mux.h</code> ) Please refer to <a href="#">Section 5.8.17.1</a> for more details.

Table 5-276. Parameters for MP4 SDK6 API **AmpMp4Mux\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> .

Table 5-277. Returns for MP4 SDK6 API **AmpMp4Mux\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

# 6 Muxer

## 6.1 Muxer: Overview

This chapter describes the APIs of the Muxer module and the Muxer handler. The primary function of the Muxer module is to handle Muxer pipes. Each Muxer pipe includes multiple formats and their media information objects. The Muxer module will process the Muxer pipes which are added to it.

The Muxer module includes the following functions:

1. Initialize the Muxer module
2. Create a Muxer pipe
3. Delete a Muxer pipe
4. Add a Muxer pipe to Muxer
5. Remove a Muxer pipe from Muxer
6. Other Muxer related functions

## 6.2 Muxer: List of APIs

- [AmpMuxer\\_Add](#)
- [AmpMuxer\\_Create](#)
- [AmpMuxer\\_Delete](#)
- [AmpMuxer\\_GetDefaultCfg](#)
- [AmpMuxer\\_GetDefaultImageInfoCfg](#)
- [AmpMuxer\\_GetDefaultMovieInfoCfg](#)
- [AmpMuxer\\_GetDefaultSoundInfoCfg](#)
- [AmpMuxer\\_GetInitDefaultCfg](#)
- [AmpMuxer\\_GetRequiredBufferSize](#)
- [AmpMuxer\\_HasEOS](#)
- [AmpMuxer\\_Init](#)
- [AmpMuxer\\_InitImageInfo](#)
- [AmpMuxer\\_InitMovieInfo](#)
- [AmpMuxer\\_InitSoundInfo](#)
- [AmpMuxer\\_LockPipe](#)
- [AmpMuxer\\_OnDataReady](#)
- [AmpMuxer\\_OnEOS](#)
- [AmpMuxer\\_Remove](#)
- [AmpMuxer\\_SetDelayTime](#)
- [AmpMuxer\\_SetMaxDuration](#)
- [AmpMuxer\\_SetProcParam](#)
- [AmpMuxer\\_Start](#)
- [AmpMuxer\\_Stop](#)
- [AmpMuxer\\_UnlockPipe](#)
- [AmpMuxer\\_WaitComplete](#)

## 6.2.1 AmpMuxer\_Add

### API Syntax:

**AmpMuxer\_Add** (AMP\_MUXER\_PIPE\_HDLR\_s \* pipe)

### Function Description:

- This function is used to add a Muxer pipe to the Muxer module.

### Parameters:

Type	Parameter	Description
AMP_MUXER_PIPE_HDLR_s *	<b>pipe</b>	The Muxer pipe being added. Please refer to <a href="#">Section 6.2.1.1</a> below for more details.

Table 6-1. Parameters for Muxer SDK6 API **AmpMuxer\_Add()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-2. Returns for Muxer SDK6 API **AmpMuxer\_Add()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 6.2.1.1 AmpMuxer\_Add > AMP\_MUXER\_PIPE\_HDLR\_s

Type	Field	Description
AMP_MUX_FORMAT_HDLR_s* [AMP_MUXER_MAX_FORMAT_PER_PIPE]	<b>Format</b>	Format handlers in a pipe (Please refer to <a href="#">Section 5.3.2.1</a> )
UINT8	<b>FormatCount</b>	The number of Format handlers in a pipe

Table 6-3. Definition of **AMP\_MUXER\_PIPE\_HDLR\_s** for Muxer SDK6 API **AmpMuxer\_Add()**.

## 6.2.2 AmpMuxer\_Create

### API Syntax:

**AmpMuxer\_Create** (AMP\_MUXER\_PIPE\_CFG\_s \* config, AMP\_MUXER\_PIPE\_HDLR\_s \*\* pipe)

### Function Description:

- This function is used to create a Muxer pipe.

### Parameters:

Type	Parameter	Description
AMP_MUXER_PIPE_CFG_s *	<b>config</b>	The configuration used to create a Muxer pipe. Please refer to <a href="#">Section 6.2.2.1</a> for more details.
AMP_MUXER_PIPE_HDLR_s **	<b>pipe</b>	The created pipe. Please refer to <a href="#">Section 6.2.1.1</a> for definition.

Table 6-4. Parameters for Muxer SDK6 API **AmpMuxer\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-5. Returns for Muxer SDK6 API **AmpMuxer\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### Note:

Media types of Muxer pipes indicate the kinds of media that will be processed. An image type cannot appear concurrently with any other types; however, movie and sound types can appear with each other.

### 6.2.2.1 AmpNuxer\_Create > AMP\_MUXER\_PIPE\_CFG\_s

Type	Field	Description
AMP_MUX_FORMAT_HDLR_s * [AMP_MUXER_MAX_FORMAT_PER_PIPE]	<b>Format</b>	Format handlers in a pipe (Please refer to <a href="#">Section 5.3.2.1</a> for more details)
AMP_MEDIA_INFO_s * [AMP_MUXER_MAX_FORMAT_PER_PIPE]	<b>Media</b>	Media information objects in a pipe. Please refer to <a href="#">Section 5.4.1.4</a> for more details.
UINT32	<b>TaskPriority</b>	The task priority of a Muxer pipe if NewTask is TRUE (The default value is the same as the one of Muxer)
AMP_CALLBACK_f	<b>OnEvent</b>	The callback function for handling Muxer events
UINT32	<b>ProcParam</b>	The process parameters of a pipe (In muxing a movie and sound, the value means process duration. No use in muxing an image)
UINT32	<b>DelayTime</b>	The time to delay muxing (ms, for movie and sound)
UINT32	<b>MaxDuration</b>	The maximum duration of the media (ms, for movie and sound)
UINT64	<b>MaxSize</b>	The maximum size of the media (for movie and sound)
UINT8	<b>FormatCount</b>	The number of the Format handlers in a pipe
BOOL8	<b>NewTask</b>	The flag indicating that the pipe will run in a new task (for realtime muxing)

Table 6-6. Definition of AMP\_MUXER\_PIPE\_CFG\_s for Muxer SDK6 API AmpMuxer\_Create().

### 6.2.3 AmpMuxer\_Delete

#### API Syntax:

**AmpMuxer\_Delete** (AMP\_MUXER\_PIPE\_HDLR\_s \* pipe)

#### Function Description:

- This function is used to delete a Muxer pipe.

#### Parameters:

Type	Parameter	Description
AMP_MUXER_PIPE_HDLR_s *	<b>pipe</b>	The Muxer pipe being deleted. Please refer to <a href="#">Section 6.2.1.1</a> for more details.

Table 6-7. Parameters for Muxer SDK6 API **AmpMuxer\_Delete()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-8. Returns for Muxer SDK6 API **AmpMuxer\_Delete()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 6.2.4 AmpMuxer\_GetDefaultCfg

### API Syntax:

**AmpMuxer\_GetDefaultCfg** (AMP\_MUXER\_PIPE\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing Muxer pipes.

### Parameters:

Type	Parameter	Description
AMP_MUXER_PIPE_CFG_s *	<b>config</b>	The returned configuration. Please refer to <a href="#">Section 6.2.2.1</a> for more details.

Table 6-9. Parameters for Muxer SDK6 API **AmpMuxer\_GetDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-10. Returns for Muxer SDK6 API **AmpMuxer\_GetDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 6.2.5 AmpMuxer\_GetDefaultImageInfoCfg

### API Syntax:

**AmpMuxer\_GetDefaultImageInfoCfg** (AMP\_MUX\_IMAGE\_INFO\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration of an image information object.

### Parameters:

Type	Parameter	Description
AMP_MUX_IMAGE_INFO_CFG_s *	<b>config</b>	The returned configuration. Please refer to <a href="#">Section 6.2.5.1</a> for more details.

Table 6-11. Parameters for Muxer SDK6 API **AmpMuxer\_GetDefaultImageInfoCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-12. Returns for Muxer SDK6 API **AmpMuxer\_GetDefaultImageInfoCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 6.2.5.1 AmpMuxer\_GetDefaultImageInfoCfg > AMP\_MUX\_IMAGE\_INFO\_CFG\_s

Type	Field	Description
AMP_FIFO_HDLR_s *	<b>Fifo</b>	The FIFO handler of image frames (Each Image information object has an individual FIFO handler) (Please refer to <a href="#">Section 3.2.6.2</a> )
UINT8 *	<b>BufferBase</b>	The start address of a FIFO buffer (Users push data into a FIFO; the FIFO will write the data to its buffer)
UINT8 *	<b>BufferLimit</b>	The end address of a FIFO buffer (FIFO size = FIFO buffer limit - FIFO buffer base)
AMP_MUX_IMAGE_FRAME_INFO_CFG_s [AMP_FORMAT_MAX_FRAME_PER_IMAGE]	<b>Frame</b>	The frames in an Image information object (Please refer to <a href="#">Section 6.2.5.2</a> for more details)
UINT8	<b>UsedFrame</b>	The number of frames stored in an image file
UINT8	<b>TotalFrame</b>	The total number of frames that this image would reference

Table 6-13. Definition of AMP\_MUX\_IMAGE\_INFO\_CFG\_s for Muxer SDK6 API AmpMuxer\_GetDefaultImageInfoCfg().

### 6.2.5.2 AmpMuxer\_GetDefaultImageInfoCfg > AMP\_MUX\_IMAGE\_FRAME\_INFO\_CFG\_s

Type	Field	Description
UINT32	<b>SeqNum</b>	The sequence number of an image
UINT32	<b>Type</b>	Image type
UINT16	<b>Width</b>	Image width
UINT16	<b>Height</b>	Image height
EXIF_INFO_s	<b>ExifInfo</b>	EXIF information (See EXIF_INFO_s)

Table 6-14. Definition of AMP\_MUX\_IMAGE\_FRAME\_INFO\_CFG\_s for Muxer SDK6 API AmpMuxer\_GetDefaultImageInfoCfg().

## 6.2.6 AmpMuxer\_GetDefaultMovieInfoCfg

### API Syntax:

**AmpMuxer\_GetDefaultMovieInfoCfg** (AMP\_MUX\_MOVIE\_INFO\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration of a Movie information object.

### Parameters:

Type	Parameter	Description
AMP_MUX_MOVIE_INFO_CFG_s *	<b>config</b>	The returned configuration. Please refer to <a href="#">Section 6.2.6.1</a> for more details.

Table 6-15. Parameters for Muxer SDK6 API **AmpMuxer\_GetDefaultMovieInfoCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-16. Returns for MuxerSDK6 API **AmpMuxer\_GetDefaultMovieInfoCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 6.2.6.1 AmpMuxer\_GetDefaultMovieInfoCfg > AMP\_MUX\_MOVIE\_INFO\_CFG\_s

Type	Field	Description
AMP_MUX_MEDIA_TRACK_CFG_s [AMP_FORMAT_MAX_TRACK_PER_MEDIA]	<b>Track</b>	Track configurations (Please refer to <a href="#">Section 6.2.6.2</a> for more details)
UINT8	<b>TrackCount</b>	The number of tracks in a Movie information object

Table 6-17. Definition of **AMP\_MUX\_MOVIE\_INFO\_CFG\_s** for Muxer SDK6 API **AmpMuxer\_GetDefaultMovieInfoCfg()**.

### 6.2.6.2 AMP\_MUX\_MOVIE\_INFO\_CFG\_s > AMP\_MUX\_MEDIA\_TRACK\_CFG\_s

Type	Field	Description
UINT32	<b>MediaId</b>	The media type of a track (The ID is a media ID. Please refer to AMP_FORMAT_MID_e)
UINT32	<b>TimeScale</b>	The ticks per second
UINT32	<b>TimePerFrame</b>	The ticks per frame
UINT32	<b>InitDelay</b>	Initial delay time of a track (ms)
AMP_FIFO_HDLR_s *	<b>Fifo</b>	The FIFO handler of a track (Each track has an individual FIFO handler) (Please refer to <a href="#">Section 3.2.6.2</a> )
UINT8*	<b>BufferBase</b>	The start address of a FIFO buffer (Users push data into a FIFO; the FIFO will write the data to its buffer)
UINT8*	<b>BufferLimit</b>	The end address of a FIFO buffer (FIFO size = FIFO buffer limit - FIFO buffer base)
union {AMP_MUX_VIDEO_TRACK_CFG_s Video; AMP_MUX_AUDIO_TRACK_CFG_s Audio; AMP_MUX_TEXT_TRACK_CFG_s Text;}	<b>Info</b>	Video: The information of a video track (Please refer to <a href="#">Section 6.2.6.3</a> )
		Audio: The information of an audio track (Please refer to <a href="#">Section 6.2.6.4</a> )
		Text: The information of a text track (Please refer to <a href="#">Section 6.2.6.5</a> )
UINT8	<b>TrackType</b>	Track type (Please refer to AMP_MEDIA_TRACK_TYPE_e.)

Table 6-18. Definition of **AMP\_MUX\_MEDIA\_TRACK\_CFG\_s** for Muxer SDK6 API **AmpMuxer\_GetDefaultMovieInfoCfg()**.

### 6.2.6.3 AMP\_MUX\_MEDIA\_TRACK\_CFG\_s > AMP\_MUX\_VIDEO\_TRACK\_CFG\_s

Type	Field	Description
UINT32	<b>GOPSize</b>	The number of pictures between IDR pictures
UINT32	<b>CodecTimeScale</b>	The time scale of the codec
UINT16	<b>Width</b>	Picture width
UINT16	<b>Height</b>	Picture height
UINT16	<b>M</b>	The number of pictures between reference pictures (IDR, I, P)
UINT16	<b>N</b>	The number of pictures between I pictures
BOOL8	<b>IsDefault</b>	The flag indicating the track is the default video track
UINT8	<b>Mode</b>	The picture mode of a video (It has progressive and interlaced mode. Interlaced mode has Field Per Sample and Frame Per Sample. See AMP_VIDEO_MODE_s.)
BOOL8	<b>ClosedGOP</b>	The flag indicating if a GOP structure is closed (The sequence pattern of a closed GOP structure is IPBBPBB, and the one of an open GOP structure is IBBPBB. If the functions of resuming or suto splitting of a video are enabled, the value is always false, open GOP.)

Table 6-19. Definition of **AMP\_MUX\_VIDEO\_TRACK\_CFG\_s** for Muxer SDK6 API **AmpMuxer\_GetDefaultMovieInfoCfg()**.

#### 6.2.6.4 AMP\_MUX\_MEDIA\_TRACK\_CFG\_s > AMP\_MUX\_AUDIO\_TRACK\_CFG\_s

Type	Field	Description
UINT32	<b>SampleRate</b>	The sample rate (Hz) of an audio track
BOOL8	<b>IsDefault</b>	The flag indicating the track is the default audio track
UINT8	<b>Channels</b>	The number of audio channels
UINT8	<b>BitsPerSample</b>	Bits per audio sample (e.g. 8 bits and 16 bits)

Table 6-20. Definition of **AMP\_MUX\_AUDIO\_TRACK\_CFG\_s** for Muxer SDK6 API **AmpMuxer\_GetDefaultMovieInfo-Cfg()**.

#### 6.2.6.5 AMP\_MUX\_MEDIA\_TRACK\_CFG\_s > AMP\_MUX\_TEXT\_TRACK\_CFG\_s

Type	Field	Description
BOOL8	<b>IsDefault</b>	The flag indicating the track is the default text track

Table 6-21. Definition of **AMP\_MUX\_TEXT\_TRACK\_CFG\_s** for Muxer SDK6 API **AmpMuxer\_GetDefaultMovieInfo-Cfg()**.

Confidential  
For HAOTEK ONLY

## 6.2.7 AmpMuxer\_GetDefaultSoundInfoCfg

### API Syntax:

**AmpMuxer\_GetDefaultSoundInfoCfg** (AMP\_MUX\_SOUND\_INFO\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration of a Sound information object.

### Parameters:

Type	Parameter	Description
AMP_MUX_SOUND_INFO_CFG_s *	<b>config</b>	The returned configuration. Please refer to <a href="#">Section 6.2.7.1</a> for more details.

Table 6-22. Parameters for Muxer SDK6 API **AmpMuxer\_GetDefaultSoundInfoCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-23. Returns for Muxer SDK6 API **AmpMuxer\_GetDefaultSoundInfoCfg()**.

### Example

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 6.2.7.1 AmpMuxer\_GetDefaultSoundInfoCfg > AMP\_MUX\_SOUND\_INFO\_CFG\_s

Type	Field	Description
AMP_MUX_MEDIA_TRACK_CFG_s [AMP_FORMAT_MAX_TRACK_PER_MEDIA]	<b>Track</b>	Track configurations (Please refer to <a href="#">Section 6.2.6.2</a> for more details)
UINT8	<b>TrackCount</b>	The number of tracks in a Sound information object

Table 6-24. Definition of **AMP\_MUX\_SOUND\_INFO\_CFG\_s** for Muxer SDK6 API **AmpMuxer\_GetDefaultSoundInfoCfg()**.

## 6.2.8 AmpMuxer\_GetInitDefaultCfg

### API Syntax:

**AmpMuxer\_GetInitDefaultCfg** (AMP\_MUXER\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the Muxer module.

### Parameters:

Type	Parameter	Description
AMP_MUXER_INIT_CFG_s *	<b>config</b>	The returned configuration. (Please refer to <a href="#">Section 6.2.8.1</a> for more details)

Table 6-25. Parameters for Muxer SDK6 API **AmpMuxer\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-26. Returns for Muxer SDK6 API **AmpMuxer\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 6.2.8.1 AmpMuxer\_GetInitDefaultCfg > AMP\_MUXER\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the Muxer module
UINT32	<b>BufferSize</b>	The size of the work buffer
AMP_TASK_INFO_s	<b>TaskInfo</b>	The information of a Muxer task (Please refer to <a href="#">Section 2.2.13.2</a> )
UINT8	<b>MaxPipe</b>	The maximum number of pipes held in the Muxer module
UINT8	<b>MaxTask</b>	The maximum number of tasks held in the Muxer module

Table 6-27. Definition of **AMP\_MUXER\_INIT\_CFG\_s** for Muxer SDK6 API **AmpMuxer\_GetInitDefaultCfg()**.

## 6.2.9 AmpMuxer\_GetRequiredBufferSize

### API Syntax:

**AmpMuxer\_GetRequiredBufferSize** (UINT8 maxPipe, UINT8 maxTask, UINT32 stackSize)

### Function Description:

- This function is used to get the required buffer size for initializing the Muxer module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxPipe</b>	The maximum number of Muxer pipes
UINT8	<b>maxTask</b>	The maximum number of Muxer tasks
UINT32	<b>stackSize</b>	The stack size of each task (byte)

Table 6-28. Parameters for Muxer SDK6 API **AmpMuxer\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size.

Table 6-29. Returns for Muxer SDK6 API **AmpMuxer\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

None



## 6.2.10 AmpMuxer\_HasEOS

### API Syntax:

**AmpMuxer\_HasEOS** (AMP\_FIFO\_HDLR\_s \*fifo)

### Function Description:

- This function is used to check if a FIFO has an EOS frame.

### Parameters:

Type	Parameter	Description
AMP_FIFO_HDLR_s *	<b>fifo</b>	The handler of a FIFO being checked. (Please refer to <a href="#">Section 3.2.6.2</a> )

Table 6-30. Parameters for Muxer SDK6 API **AmpMuxer\_HasEOS()**.

### Returns:

Return	Description
True	Return true if the FIFO has an EOS frame
False	Return false if the FIFO has no EOS frame

Table 6-31. Returns for Muxer SDK6 API **AmpMuxer\_HasEOS()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 6.2.11 AmpMuxer\_Init

### API Syntax:

**AmpMuxer\_Init** (AMP\_MUXER\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initialize the Muxer module.

### Parameters:

Type	Parameter	Description
AMP_MUXER_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the Muxer module. (Please refer to <a href="#">Section 6.2.8.1</a> for more details)

Table 6-32. Parameters for Muxer SDK6 API **AmpMuxer\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-33. Returns for Muxer SDK6 API **AmpMuxer\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 6.2.12 AmpMuxer\_InitImageInfo

### API Syntax:

**AmpMuxer\_InitImageInfo** (AMP\_IMAGE\_INFO\_s \* image, AMP\_MUX\_IMAGE\_INFO\_CFG\_s \* config)

### Function Description:

- This function is used to initialize an Image information object.

### Parameters:

Type	Parameter	Description
AMP_IMAGE_INFO_s *	<b>image</b>	The image information object being initialized. Please refer to <a href="#">Section 5.2.1.1</a> for more details.
AMP_MUX_IMAGE_INFO_CFG_s *	<b>config</b>	The configuration used to initialize an Image information object. Please refer to <a href="#">Section 6.2.5.1</a> for more details.

Table 6-34. Parameters for Muxer SDK6 API **AmpMuxer\_InitImageInfo()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-35. Returns for Muxer SDK6 API **AmpMuxer\_InitImageInfo()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 6.2.13 AmpMuxer\_InitMovieInfo

#### API Syntax:

**AmpMuxer\_InitMovieInfo** (AMP\_MOVIE\_INFO\_s \* movie, AMP\_MUX\_MOVIE\_INFO\_CFG\_s \* config)

#### Function Description:

- This function is used to initialize a Movie information object.

#### Parameters:

Type	Parameter	Description
AMP_MOVIE_INFO_s *	<b>movie</b>	The Movie information object being initialized. Please refer to <a href="#">Section 5.2.2.1</a> for more details.
AMP_MUX_MOVIE_INFO_CFG_s *	<b>config</b>	The configuration used to initialize a Movie information object. Please refer to <a href="#">Section 6.2.6.1</a> for more details.

Table 6-36. Parameters for Muxer SDK6 API **AmpMuxer\_InitMovieInfo()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-37. Returns for Muxer SDK6 API **AmpMuxer\_InitMovieInfo()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 6.2.14 AmpMuxer\_InitSoundInfo

### API Syntax:

**AmpMuxer\_InitSoundInfo** (AMP\_SOUND\_INFO\_s \* sound, AMP\_MUX\_SOUND\_INFO\_CFG\_s \* config)

### Function Description:

- This function is used to initialize a Sound information object.

### Parameters:

Type	Parameter	Description
AMP_SOUND_INFO_s *	<b>sound</b>	The Sound information object being initialized. Please refer to <a href="#">Section 5.2.3.1</a> for more details.
AMP_MUX_SOUND_INFO_CFG_s *	<b>config</b>	The configuration used to initialize a Sound information object. Please refer to <a href="#">Section 6.2.7.1</a> for more details.

Table 6-38. Parameters for Muxer SDK6 API **AmpMuxer\_InitSoundInfo()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-39. Returns for Muxer SDK6 API **AmpMuxer\_InitSoundInfo()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 6.2.15 AmpMuxer\_LockPipe

### API Syntax:

**AmpMuxer\_LockPipe** (AMP\_MUXER\_PIPE\_HDLR\_s \* pipe)

### Function Description:

- This function is used to lock a Muxer pipe.

### Parameters:

Type	Parameter	Description
AMP_MUXER_PIPE_HDLR_s *	<b>pipe</b>	The Muxer pipe being locked. Please refer to <a href="#">Section 6.2.1.1</a> for more details.

Table 6-40. Parameters for Muxer SDK6 API **AmpMuxer\_LockPipe()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-41. Returns for Muxer SDK6 API **AmpMuxer\_LockPipe()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 6.2.16 AmpMuxer\_OnDataReady

### API Syntax:

**AmpMuxer\_OnDataReady** (AMP\_FIFO\_HDLR\_s \* fifo)

### Function Description:

- This function is used to notify Muxer of new frames being available in a FIFO.

### Parameters:

Type	Parameter	Description
AMP_FIFO_HDLR_s *	<b>fifo</b>	The handler of a FIFO holding new frames. (Please refer to <a href="#">Section 3.2.6.2</a> )

Table 6-42. Parameters for Muxer SDK6 API **AmpMuxer\_OnDataReady()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-43. Returns for Muxer SDK6 API **AmpMuxer\_OnDataReady()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 6.2.17 AmpMuxer\_OnEOS

### API Syntax:

**AmpMuxer\_OnEOS** ( AMP\_FIFO\_HDLR\_s \* fifo)

### Function Description:

- This function is used to notify Muxer that an EOS frame has appeared in a FIFO.

### Parameters:

Type	Parameter	Description
AMP_FIFO_HDLR_s *	<b>fifo</b>	The handler of a FIFO in which an EOS frame appears. (Please refer to <a href="#">Section 3.2.6.2</a> )

Table 6-44. Parameters for Muxer SDK6 API **AmpMuxer\_OnEOS()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-45. Returns for Muxer SDK6 API **AmpMuxer\_OnEOS()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 6.2.18 AmpMuxer\_Remove

### API Syntax:

**AmpMuxer\_Remove** (AMP\_MUXER\_PIPE\_HDLR\_s \* pipe)

### Function Description:

- This function is used to remove a Muxer pipe.

### Parameters:

Type	Parameter	Description
AMP_MUXER_PIPE_HDLR_s *	<b>pipe</b>	The Muxer pipe being removed. Please refer to <a href="#">Section 6.2.1.1</a> for more details.

Table 6-46. Parameters for Muxer SDK6 API **AmpMuxer\_Remove()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-47. Returns for Muxer SDK6 API **AmpMuxer\_Remove()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 6.2.19 AmpMuxer\_SetDelayTime

### API Syntax:

**AmpMuxer\_SetDelayTime** (AMP\_MUXER\_PIPE\_HDLR\_s \* pipe, UINT32 delayTime)

### Function Description:

- This function is used to set the delay time of a Muxer pipe.

### Parameters:

Type	Parameter	Description
AMP_MUXER_PIPE_HDLR_s *	<b>pipe</b>	The pipe applying the delay time. Please refer to <a href="#">Section 6.2.1.1</a> for more details.
UINT32	<b>delayTime</b>	The delay time (ms)

Table 6-48. Parameters for Muxer SDK6 API **AmpMuxer\_SetDelayTime()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-49. Returns for Muxer SDK6 API **AmpMuxer\_SetDelayTime()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 6.2.20 AmpMuxer\_SetMaxDuration

### API Syntax:

**AmpMuxer\_SetMaxDuration** (AMP\_MUXER\_PIPE\_HDLR\_s \* pipe, UINT32 maxDuration)

### Function Description:

- This function is used to set the maximum duration of a Muxer pipe.

### Parameters:

Type	Parameter	Description
AMP_MUXER_PIPE_HDLR_s *	<b>pipe</b>	The pipe applying the maximum duration. Please refer to <a href="#">Section 6.2.1.1</a> for more details.
UINT32	<b>maxDuration</b>	The maximum duration (ms)

Table 6-50. Parameters for Muxer SDK6 API **AmpMuxer\_SetMaxDuration()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-51. Returns for Muxer SDK6 API **AmpMuxer\_SetMaxDuration()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 6.2.21 AmpMuxer\_SetProcParam

### API Syntax:

**AmpMuxer\_SetProcParam** (AMP\_MUXER\_PIPE\_HDLR\_s \* pipe, UINT32 procParam)

### Function Description:

- This function is used to set the process parameters of a Muxer pipe.

### Parameters:

Type	Parameter	Description
AMP_MUXER_PIPE_HDLR_s *	<b>pipe</b>	The pipe applying the parameter. Please refer to <a href="#">Section 6.2.1.1</a> for more details.
UINT32	<b>procParam</b>	The process parameter

Table 6-52. Parameters for Muxer SDK6 API **AmpMuxer\_SetProcParam()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-53. Returns for Muxer SDK6 API **AmpMuxer\_SetProcParam()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 6.2.22 AmpMuxer\_Start

### API Syntax:

**AmpMuxer\_Start** (AMP\_MUXER\_PIPE\_HDLR\_s \* pipe, UINT32 timeOut)

### Function Description:

- This function is used to start a Muxer pipe. After invoking the function, the start will be postponed until the first frame is available.

### Parameters:

Type	Parameter	Description
AMP_MUXER_PIPE_HDLR_s *	<b>pipe</b>	The Muxer pipe being started. Please refer to <a href="#">Section 6.2.1.1</a> for more details.
UINT32	<b>timeOut</b>	The timeout value

Table 6-54. Parameters for Muxer SDK6 API **AmpMuxer\_Start()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-55. Returns for Muxer SDK6 API **AmpMuxer\_Start()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 6.2.23 AmpMuxer\_Stop

#### API Syntax:

**AmpMuxer\_Stop** (AMP\_MUXER\_PIPE\_HDLR\_s \* pipe)

#### Function Description:

- This function is used to stop a Muxer pipe. Because a clip would be ended at the last available IDR/I/P, after invoking the function, the stop will be postponed until all frames are processed.

#### Parameters:

Type	Parameter	Description
AMP_MUXER_PIPE_HDLR_s *	<b>pipe</b>	The Muxer pipe being stopped. Please refer to <a href="#">Section 6.2.1.1</a> for more details.

Table 6-56. Parameters for Muxer SDK6 API **AmpMuxer\_Stop()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-57. Returns for Muxer SDK6 API **AmpMuxer\_Stop()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 6.2.24 AmpMuxer\_UnlockPipe

### API Syntax:

**AmpMuxer\_UnlockPipe** (AMP\_MUXER\_PIPE\_HDLR\_s \* pipe)

### Function Description:

- This function is used to unlock the Muxer pipe.

### Parameters:

Type	Parameter	Description
AMP_MUXER_PIPE_HDLR_s *	<b>pipe</b>	The Muxer pipe being unlocked. Please refer to <a href="#">Section 6.2.1.1</a> for more details.

Table 6-58. Parameters for Muxer SDK6 API **AmpMuxer\_UnlockPipe()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-59. Returns for Muxer SDK6 API **AmpMuxer\_UnlockPipe()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 6.2.25 AmpMuxer\_WaitComplete

### API Syntax:

**AmpMuxer\_WaitComplete** (AMP\_MUXER\_PIPE\_HDLR\_s \* pipe, UINT32 timeOut)

### Function Description:

- This function is used to poll the status of a pipe to check whether its life cycle is complete.

### Parameters:

Type	Parameter	Description
AMP_MUXER_PIPE_HDLR_s *	<b>pipe</b>	The pipe being polled. Please refer to <a href="#">Section 6.2.1.1</a> for more details.
UINT32	<b>timeOut</b>	The polling interval (ms)

Table 6-60. Parameters for Muxer SDK6 API **AmpMuxer\_WaitComplete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 6-61. Returns for Muxer SDK6 API **AmpMuxer\_WaitComplete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



# 7 Index

## 7.1 Index: Overview

The index module provides index utilities for the Format module. When muxing/editing, some muxing/editing formats need temporal spaces to keep intermediate data for indexing the media. Index module provides the API and interface to access the index spaces.

## 7.2 Index: List of APIs

- [AmpIndex\\_AllocateBuffer](#)
- [AmpIndex\\_GetInitDefaultCfg](#)
- [AmpIndex\\_GetRequiredBufferSize](#)
- [AmpIndex\\_Init](#)
- [AmpIndex\\_ReleaseBuffer](#)
- [AmpMemIdx\\_Create](#)
- [AmpMemIdx\\_Delete](#)
- [AmpMemIdx\\_GetDefaultCfg](#)
- [AmpMemIdx\\_GetInitDefaultCfg](#)
- [AmpMemIdx\\_GetRequiredBufferSize](#)
- [AmpMemIdx\\_Init](#)
- [AmpRawIdx\\_Create](#)
- [AmpRawIdx\\_Delete](#)
- [AmpRawIdx\\_GetDefaultCfg](#)
- [AmpRawIdx\\_GetInitDefaultCfg](#)
- [AmpRawIdx\\_GetRequiredBufSize](#)
- [AmpRawIdx\\_Init](#)
- [AmpTemplIdx\\_Create](#)
- [AmpTemplIdx\\_Delete](#)
- [AmpTemplIdx\\_GetDefaultCfg](#)
- [AmpTemplIdx\\_GetInitDefaultCfg](#)
- [AmpTemplIdx\\_GetRequiredBufferSize](#)
- [AmpTemplIdx\\_Init](#)

## 7.2.1 **AmplIndex\_AllocateBuffer**

### API Syntax:

**AmplIndex\_AllocateBuffer** (UINT32 size)

### Function Description:

- This function is used to allocate an index buffer.

### Parameters:

Type	Parameter	Description
UINT32	size	The size of an index I/O buffer.

Table 7-1. Parameters for Index SDK6 API **AmplIndex\_AllocateBuffer()**.

### Returns:

Return	Description
The index buffer address	The index buffer address

Table 7-2. Returns for Index SDK6 API **AmplIndex\_AllocateBuffer()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 7.2.2 AmplIndex\_GetInitDefaultCfg

### API Syntax:

**AmplIndex\_GetInitDefaultCfg** (AMP\_INDEX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the Index module.

### Parameters:

Type	Parameter	Description
AMP_INDEX_INIT_CFG_s *	<b>config</b>	The returned configuration. Please refer to <a href="#">Section 7.2.2.1</a> for more details.

Table 7-3. Parameters for Index SDK6 API **AmplIndex\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-4. Returns for Index SDK6 API **AmplIndex\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 7.2.2.1 AmplIndex\_GetInitDefaultCfg > AMP\_INDEX\_INIT\_CFG\_s

Type	Field	Description
void *	<b>Buffer</b>	The work buffer of the Index module
UINT32	<b>BufferSize</b>	The size of the work buffer
UINT8	<b>MaxHdlr</b>	The maximum number of index handlers held in Index

Table 7-5. Definition of **AMP\_INDEX\_INIT\_CFG\_s** for Index SDK6 API **AmplIndex\_GetInitDefaultCfg()**.

### 7.2.3 AmplIndex\_GetRequiredBufferSize

#### API Syntax:

**AmplIndex\_GetRequiredBufferSize** (UINT8 maxHdlr, UINT32 memSize)

#### Function Description:

- This function is used to get the required buffer size for initializing the Index module.

#### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of index handlers
UINT32	<b>memSize</b>	The memory size of the index I/O buffer

Table 7-6. Parameters for Index SDK6 API **AmplIndex\_GetRequiredBufferSize()**.

#### Returns:

Return	Description
The required buffer size	The required buffer size

Table 7-7. Returns for Index SDK6 API **AmplIndex\_GetRequiredBufferSize()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 7.2.4 AmplIndex\_Init

### API Syntax:

**AmplIndex\_Init** (AMP\_INDEX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initialize the Index module.

### Parameters:

Type	Parameter	Description
AMP_INDEX_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the Index module. Please refer to <a href="#">Section 7.2.2.1</a> for more details.

Table 7-8. Parameters for Index SDK6 API **AmplIndex\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-9. Returns for Index SDK6 API **AmplIndex\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 7.2.5 `AmplIndex_ReleaseBuffer`

### API Syntax:

**`AmplIndex_ReleaseBuffer`** (UINT8 \* buffer)

### Function Description:

- This function is used to release an index buffer.

### Parameters:

Type	Parameter	Description
UINT8 *	<b>buffer</b>	The index I/O buffer being released

Table 7-10. Parameters for Index SDK6 API **`AmplIndex_ReleaseBuffer()`**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-11. Returns for Index SDK6 API **`AmplIndex_ReleaseBuffer()`**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 7.2.6 AmpMemIdx\_Create

### API Syntax:

**AmpMemIdx\_Create** (AMP\_MEM\_IDX\_CFG\_s \* config, AMP\_INDEX\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create an MEM index handler.

### Parameters:

Type	Parameter	Description
AMP_MEM_IDX_CFG_s *	<b>config</b>	The configuration used to create an MEM index handler index. Please refer to <a href="#">Section 7.2.6.1</a> for more details.
AMP_INDEX_HDLR_s **	<b>hdlr</b>	The returned MEM index handler. Please refer to <a href="#">Section 7.2.6.2</a> for more details.

Table 7-12. Parameters for Index SDK6 API **AmpMemIdx\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-13. Returns for Index SDK6 API **AmpMemIdx\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 7.2.6.1 AmpMemIdx\_Create > AMP\_MEM\_IDX\_CFG\_s

Type	Field	Description
UINT8[4]	<b>Resv</b>	Reserved

Table 7-14. Definition of **AMP\_MEM\_IDX\_CFG\_s** for Index SDK6 API **AmpMemIdx\_Create()**.

### 7.2.6.2 AmpMemIdx\_Create > AMP\_INDEX\_HDLR\_s

Type	Field	Description
AMP_INDEX_s *	<b>Func</b>	The functions used to operate an index file (Please refer to <a href="#">Section 7.2.6.3</a> )

Table 7-15. Definition of **AMP\_INDEX\_HDLR\_s** for Index SDK6 API **AmpMemIdx\_Create()**.

### 7.2.6.3 AmpMemIdx\_Create > AMP\_INDEX\_s

Type	Field	Description
int (*)(AMP_INDEX_HDLR_s *, char *, INT32)	<b>Open</b>	The interface to open an Index file (*pHdlr, *Url, Size)
int (*)(AMP_INDEX_HDLR_s *, BOOL8)	<b>Close</b>	The interface to close an Index file (*pHdlr, bRemove)
int (*)(AMP_INDEX_HDLR_s *, UINT32, UINT32, UINT8 *)	<b>Read</b>	The interface to read data from an Index file (*pHdlr, nOffset, Size, *Buffer)
int (*)(AMP_INDEX_HDLR_s *, UINT32, UINT32, UINT8 *)	<b>Write</b>	The interface to write data into an Index file (*pHdlr, nOffset, Size, *Buffer)
int (*)(AMP_INDEX_HDLR_s *)	<b>Sync</b>	The interface to flush an Index file (*pHdlr)
int (*)(AMP_INDEX_DEV_INFO_s *)	<b>Info</b>	The interface to obtain the information of a device where Index files place (*pInfo) (Please refer to <a href="#">Section 7.2.6.4</a> )
int (*)(UINT32, UINT32, UINT32)	<b>Func</b>	The interface to invoke other functions with a command code (nCmd, nParam1, nParam2)

Table 7-16. Definition of **AMP\_INDEX\_s** for Index SDK6 API **AmpMemIdx\_Create()**.

### 7.2.6.4 AMP\_INDEX\_s > AMP\_INDEX\_DEV\_INFO\_s

Type	Field	Description
UINT32	<b>SectorSize</b>	The size of a device sector
UINT32	<b>PageSize</b>	The size of a device page
UINT32	<b>BlockSize</b>	The size of a device block

Table 7-17. Definition of **AMP\_INDEX\_DEV\_INFO\_s** for Index SDK6 API **AmpMemIdx\_Create()**.



## 7.2.7 AmpMemIdx\_Delete

### API Syntax:

**AmpMemIdx\_Delete** (AMP\_INDEX\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete an MEM index handler.

### Parameters:

Type	Parameter	Description
AMP_INDEX_HDLR_s *	hdlr	The MEM index handler being deleted. Please refer to <a href="#">Section 7.2.6.2</a> for more details.

Table 7-18. Parameters for Index SDK6 API **AmpMemIdx\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-19. Returns for Index SDK6 API **AmpMemIdx\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 7.2.8 AmpMemIdx\_GetDefaultCfg

### API Syntax:

**AmpMemIdx\_GetDefaultCfg** (AMP\_MEM\_IDX\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for creating an MEM index handler.

### Parameters:

Type	Parameter	Description
AMP_MEM_IDX_CFG_s *	<b>config</b>	The returned configuration. Please refer to <a href="#">Section 7.2.6.1</a> for more details.

Table 7-20. Parameters for Index SDK6 API **AmpMemIdx\_GetDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-21. Returns for Index SDK6 API **AmpMemIdx\_GetDefaultCfg()**.

### Example

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 7.2.9 AmpMemIdx\_GetInitDefaultCfg

### API Syntax:

**AmpMemIdx\_GetInitDefaultCfg** (AMP\_MEM\_IDX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the MEM index module.

### Parameters:

Type	Parameter	Description
AMP_MEM_IDX_INIT_CFG_s *	<b>config</b>	The returned configuration. Please refer to <a href="#">Section 7.2.9.1</a> for more details.

Table 7-22. Parameters for Index SDK6 API **AmpMemIdx\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-23. Returns for Index SDK6 API **AmpMemIdx\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 7.2.9.1 AmpMemIdx\_GetInitDefaultCfg > AMP\_MEM\_IDX\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the MEM index module
UINT32	<b>BufferSize</b>	The size of the work buffer
UINT8	<b>MaxHdlr</b>	The maximum number of MEM index handlers held in the MEM index module

Table 7-24. Definition of **AMP\_MEM\_IDX\_INIT\_CFG\_s** for Index SDK6 API **AmpMemIdx\_GetInitDefaultCfg()**.

## 7.2.10 AmpMemIdx\_GetRequiredBufferSize

### API Syntax:

**AmpMemIdx\_GetRequiredBufferSize** (UINT8 maxHdlr, UINT32 memSize)

### Function Description:

- This function is used to get the required buffer size for initializing the MEM index module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of MEM index handlers
UINT32	<b>memSize</b>	The size of memory buffer

Table 7-25. Parameters for Index SDK6 API **AmpMemIdx\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 7-26. Returns for Index SDK6 API **AmpMemIdx\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 7.2.11 AmpMemIdx\_Init

### API Syntax:

**AmpMemIdx\_Init** (AMP\_MEM\_IDX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initialize the MEM index module.

### Parameters:

Type	Parameter	Description
AMP_MEM_IDX_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the MEM index module. Please refer to <a href="#">Section 7.2.9.1</a> for more details.

Table 7-27. Parameters for Index SDK6 API **AmpMemIdx\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-28. Returns for Index SDK6 API **AmpMemIdx\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 7.2.12 AmpRawIdx\_Create

### API Syntax:

**AmpRawIdx\_Create** (AMP\_RAW\_IDX\_CFG\_s \* config, AMP\_INDEX\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create a RAW index handler.

### Parameters:

Type	Parameter	Description
AMP_RAW_IDX_CFG_s *	<b>config</b>	The configuration used to create a Raw index handler. Please refer to <a href="#">Section 7.2.12.1</a> for more details.
AMP_INDEX_HDLR_s **	<b>hdlr</b>	The returned Raw index handler. Please refer to <a href="#">Section 7.2.6.2</a> for more details.

Table 7-29. Parameters for Index SDK6 API **AmpRawIdx\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-30. Returns for Index SDK6 API **AmpRawIdx\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 7.2.12.1 AmpRawIdx\_Create > AMP\_RAW\_IDX\_CFG\_s

Type	Field	Description
UINT8[4]	<b>Resv</b>	Reserved

Table 7-31. Definition of **AMP\_RAW\_IDX\_CFG\_s** for Index SDK6 API **AmpRawIdx\_Create()**.

### 7.2.13 AmpRawIdx\_Delete

#### API Syntax:

**AmpRawIdx\_Delete** (AMP\_INDEX\_HDLR\_s \* hdlr)

#### Function Description:

- This function is used to delete a RAW index handler.

#### Parameters:

Type	Parameter	Description
AMP_INDEX_HDLR_s *	hdlr	The Raw index handler being deleted. Please refer to <a href="#">Section 7.2.6.2</a> for more details.

Table 7-32. Parameters for Index SDK6 API **AmpRawIdx\_Delete()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-33. Returns for Index SDK6 API **AmpRawIdx\_Delete()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 7.2.14 AmpRawIdx\_GetDefaultCfg

### API Syntax:

**AmpRawIdx\_GetDefaultCfg** (AMP\_RAW\_IDX\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for creating a Raw index handler.

### Parameters:

Type	Parameter	Description
AMP_RAW_IDX_CFG_s *	<b>config</b>	The returned configuration. Please refer to <a href="#">Section 7.2.12.1</a> for more details.

Table 7-34. Parameters for Index SDK6 API **AmpRawIdx\_GetDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-35. Returns for Index SDK6 API **AmpRawIdx\_GetDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 7.2.15 AmpRawIdx\_GetInitDefaultCfg

### API Syntax:

**AmpRawIdx\_GetInitDefaultCfg** (AMP\_RAW\_IDX\_INIT\_CFG\_s \* config, UINT8 DevType)

### Function Description:

- This function is used to get the default configuration for initializing the Raw index module.

### Parameters:

Type	Parameter	Description
AMP_RAW_IDX_INIT_CFG_s *	<b>config</b>	The returned configuration. Please refer to <a href="#">Section 7.2.15.1</a> for more details.
UINT8	<b>DevType</b>	The device type (See AMP_RAW_IDX_DEV_TYPE_e in <a href="#">Section 7.2.15.4</a> )

Table 7-36. Parameters for Index SDK6 API **AmpRawIdx\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-37. Returns for Index SDK6 API **AmpRawIdx\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 7.2.15.1 AmpRawIdx\_GetInitDefaultCfg > AMP\_RAW\_IDX\_INIT\_CFG\_s

Type	Field	Description
UINT8 *	<b>Buffer</b>	The work buffer of the Raw index module
UINT32	<b>BufferSize</b>	The size of the work buffer
union {AMP_RAW_IDX_NAND_INFO_s Nand; AMP_RAW_IDX_EMMC_INFO_s Emmc;}	<b>DevInfo</b>	Nand: NAND information ( <a href="#">Section 7.2.15.2</a> ) Emmc: EMMC information ( <a href="#">Section 7.2.15.3</a> )

Type	Field	Description
UINT8	<b>DevType</b>	The device type (See AMP_RAW_IDX_DEV_TYPE_e in <a href="#">Section 7.2.15.4</a> )
UINT8	<b>PartId</b>	The partition Id
UINT8	<b>MaxHdlr</b>	The maximum number of Raw index handlers held in the Raw index module

Table 7-38. Definition of **AMP\_RAW\_IDX\_INIT\_CFG\_s** for Index SDK6 API **AmpRawIdx\_GetInitDefaultCfg()**.

### 7.2.15.2 AmpRawIdx\_GetInitDefaultCfg > AMP\_RAW\_IDX\_NAND\_INFO\_s

Type	Field	Description
AMBA_NAND_DEV_INFO_s	<b>Info</b>	NAND information
UINT8	<b>Mode</b>	NAND initiation mode (NFTL_MODE_NO_SAVE_TRL_TBL or NFTL_MODE_SAVE_TRL_TBL)

Table 7-39. Definition of **AMP\_RAW\_IDX\_NAND\_INFO\_s** for Index SDK6 API **AmpRawIdx\_GetInitDefaultCfg()**.

### 7.2.15.3 AmpRawIdx\_GetInitDefaultCfg > AMP\_RAW\_IDX\_EMMC\_INFO\_s

Type	Field	Description
UINT8[4]	<b>Resv</b>	Reserved

Table 7-40. Definition of **AMP\_RAW\_IDX\_EMMC\_INFO\_s** for Index SDK6 API **AmpRawIdx\_GetInitDefaultCfg()**.

### 7.2.15.4 AmpRawIdx\_GetInitDefaultCfg > AMP\_RAW\_IDX\_DEV\_TYPE\_e

Type	Field	Description
AMP_RAW_IDX_DEV_TYPE_NAND	<b>Nand</b>	Nand Type
AMP_RAW_IDX_DEV_TYPE_EMMC	<b>Emmc</b>	Emmc Type

Table 7-41. Definition of **AMP\_RAW\_IDX\_DEV\_TYPE\_e** for Index SDK6 API **AmpRawIdx\_GetInitDefaultCfg()**.

## 7.2.16 AmpRawIdx\_GetRequiredBufferSize

### API Syntax:

**AmpRawIdx\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the Raw index module.

### Parameters:

Type	Parameter	Description
UINT8	maxHdlr	Maximum number of RAW index handlers

Table 7-42. Parameters for Index SDK6 API **AmpRawIdx\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 7-43. Returns for Index SDK6 API **AmpRawIdx\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 7.2.17 AmpRawIdx\_Init

### API Syntax:

**AmpRawIdx\_Init** (AMP\_RAW\_IDX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initialize the Raw index module.

### Parameters:

Type	Parameter	Description
AMP_RAW_IDX_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the Raw index module. Please refer to <a href="#">Section 7.2.15.1</a> for more details.

Table 7-44. Parameters for Index SDK6 API **AmpRawIdx\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-45. Returns for Index SDK6 API **AmpRawIdx\_Init()**.

### Example:

Please refer to Unit Test document.

### See Also:

None

## 7.2.18 AmpTempldx\_Create

### API Syntax:

**AmpTempldx\_Create** (AMP\_TEMP\_IDX\_CFG\_s \* config, AMP\_INDEX\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create a Temp index handler.

### Parameters:

Type	Parameter	Description
AMP_TEMP_IDX_CFG_s *	<b>config</b>	The configuration used to create a Temp index handler. Please refer to <a href="#">Section 7.2.18.1</a> for more details.
AMP_INDEX_HDLR_s **	<b>hdlr</b>	The returned Temp index handler. Please refer to <a href="#">Section 7.2.6.2</a> for more details.

Table 7-46. Parameters for Index SDK6 API **AmpTempldx\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-47. Returns for Index SDK6 API **AmpTempldx\_Create()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 7.2.18.1 AmpTempldx\_Create > AMP\_TEMP\_IDX\_CFG\_s

Type	Field	Description
UINT8[4]	<b>Resv</b>	Reserved

Table 7-48. Definition of **AMP\_TEMP\_IDX\_CFG\_s** for Index SDK6 API **AmpTempldx\_Create()**.

## 7.2.19 AmpTempldx\_Delete

### API Syntax:

**AmpTempldx\_Delete** (AMP\_INDEX\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete a TEMP index handler.

### Parameters:

Type	Parameter	Description
AMP_INDEX_HDLR_s *	hdlr	The TEMP index handler being deleted. Please refer to <a href="#">Section 7.2.6.2</a> for more details.

Table 7-49. Parameters for Index SDK6 API **AmpTempldx\_Delete()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-50. Returns for Index SDK6 API **AmpTempldx\_Delete()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 7.2.20 AmpTempldx\_GetDefaultCfg

### API Syntax:

**AmpTempldx\_GetDefaultCfg** (AMP\_TEMP\_IDX\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for creating a Temp index handler.

### Parameters:

Type	Parameter	Description
AMP_TEMP_IDX_CFG_s *	<b>config</b>	The returned configuration. Please refer to <a href="#">Section 7.2.18.1</a> for more details.

Table 7-51. Parameters for Index SDK6 API **AmpTempldx\_GetDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-52. Returns for Index SDK6 API **AmpTempldx\_GetDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 7.2.21 AmpTempldx\_GetInitDefaultCfg

### API Syntax:

**AmpTempldx\_GetInitDefaultCfg** (AMP\_TEMP\_IDX\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the Temp index module.

### Parameters:

Type	Parameter	Description
AMP_TEMP_IDX_INIT_CFG_s *	<b>config</b>	The returned configuration. Please refer to <a href="#">Section 7.2.21.1</a> for more details.

Table 7-53. Parameters for Index SDK6 API **AmpTempldx\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-54. Returns for Index SDK6 API **AmpTempldx\_GetInitDefaultCfg()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 7.2.21.1 AmpTempldx\_GetInitDefaultCfg > AMP\_TEMP\_IDX\_INIT\_CFG\_s

Type	Field	Description
char [AMP_TEMP_IDX_MAX_FNEXT_LENGTH]	<b>Ext</b>	The extension name of an index file
UINT8 *	<b>Buffer</b>	The work buffer of the Temp index module
UINT32	<b>BufferSize</b>	The size of the work buffer
UINT8	<b>MaxHdlr</b>	The maximum number of Temp index handlers held in the Temp index module

Table 7-55. Definition of **AMP\_TEMP\_IDX\_INIT\_CFG\_s** for Index SDK6 API **AmpTempldx\_GetInitDefaultCfg()**.



## 7.2.22 AmpTempldx\_GetRequiredBufferSize

### API Syntax:

**AmpTempldx\_GetRequiredBufSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the Temp index module.

### Parameters:

Type	Parameter	Description
UINT8	maxHdlr	The maximum number of Temp index handlers.

Table 7-56. Parameters for Index SDK6 API **AmpTempldx\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 7-57. Returns for Index SDK6 API **AmpTempldx\_GetRequiredBufferSize()**.

### Example:

Please refer to Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 7.2.23 AmpTempldx\_Init

#### API Syntax:

**AmpTempldx\_Init** (AMP\_TEMP\_IDX\_INIT\_CFG\_s \* config)

#### Function Description:

- This function is used to initialize the Temp index module.

#### Parameters:

Type	Parameter	Description
AMP_TEMP_IDX_INIT_CFG_s *	<b>config</b>	The configuration used to initialize the Temp index module. Please refer to <a href="#">Section 7.2.21.1</a> for more details.

Table 7-58. Parameters for Index SDK6 API **AmpTempldx\_Init()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All other	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 7-59. Returns for Index SDK6 API **AmpTempldx\_Init()**.

#### Example:

Please refer to Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

# 8 Stream

## 8.1 Stream: Overview

Stream module provides the APIs and interfaces to access I/O streams.

The File Stream module includes the following functions:

1. Initiate the File Stream module
2. Create file streams
3. Delete file streams

## 8.2 Stream: List of APIs

This section lists the stream APIs:

- [AmpFileStream\\_Create](#)
- [AmpFileStream\\_Delete](#)
- [AmpFileStream\\_GetDefaultCfg](#)
- [AmpFileStream\\_GetInitDefaultCfg](#)
- [AmpFileStream\\_GetRequiredBufferSize](#)
- [AmpFileStream\\_Init](#)

## 8.2.1 AmpFileStream\_Create

### API Syntax:

**AmpFileStream\_Create** (AMP\_FILE\_STREAM\_CFG\_s \* config, AMP\_STREAM\_HDLR\_s \*\* hdlr)

### Function Description:

- This function is used to create a file stream handler.

### Parameters:

Type	Parameter	Description
AMP_FILE_STREAM_CFG_s*	<b>config</b>	The configuration used to create a file stream handler ( <b>AMP_FILE_STREAM_CFG_s</b> is defined in <code>File.h</code> ) Please refer to <a href="#">Section 8.2.1.1</a> for more details.
AMP_STREAM_HDLR_s**	<b>hdlr</b>	The returned handler ( <b>AMP_STREAM_HDLR_s</b> is defined in <code>Stream.h</code> ) Please refer to <a href="#">Section 8.2.1.2</a> for more details.

Table 8-1. Parameters for Stream SDK6 API **AmpFileStream\_Create()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 8-2. Returns for Stream SDK6 API **AmpFileStream\_Create()**.

### 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 AmpFileStream\_Create > AMP\_FILE\_STREAM\_CFG\_s

Type	Field	Description
UINT32	<b>Alignment</b>	The alignment of file size
UINT32	<b>BytesToSync</b>	The number of bytes to sync FAT
AMP_FILE_STREAM_ASYNC_PARAM_s	<b>AsyncParam</b>	The parameters of async mode ( <b>AMP_FILE_STREAM_ASYNC_PARAM_s</b> is defined in <code>File.h</code> ) Please refer to <a href="#">Section 8.2.1.1.1</a> for more details.
BOOL8	<b>Async</b>	The flag to enable async mode (The handler runs in async mode.)

Type	Field	Description
BOOL8	<b>LowPriority</b>	The flag to enable low priority (The handler runs in low priority mode)

Table 8-3. Definition of **AMP\_FILE\_STREAM\_CFG\_s** for Stream SDK6 API **AmpFileStream\_Create()**.

#### 8.2.1.1.1 AMP\_FILE\_STREAM\_CFG\_s > AMP\_FILE\_STREAM\_ASYNC\_PARAM\_s

Type	Field	Description
UINT8	<b>MaxBank</b>	The maximum number of banks that the handler can use

Table 8-4. Definition of **AMP\_FILE\_STREAM\_ASYNC\_PARAM\_s** for Stream SDK6 API **AmpFileStream\_Create()**.

#### 8.2.1.2 AmpFileStream\_Create > AMP\_STREAM\_HDLR\_s

Type	Field	Description
AMP_STREAM_s*	<b>Func</b>	Stream interface ( <b>AMP_STREAM_s</b> is defined in <code>Stream.h</code> ). Please refer to <a href="#">Section 8.2.1.2.1</a> for more details.

Table 8-5. Definition of **AMP\_STREAM\_HDLR\_s** for Stream SDK6 API **AmpFileStream\_Create()**.

#### 8.2.1.2.1 AMP\_FILE\_STREAM\_HDLR\_s > AMP\_STREAM\_s

Type	Field	Description
int(*)( <b>AMP_STREAM_HDLR_s*</b> , char*, <b>UINT32</b> )	<b>Open</b>	The interface to open a stream handler
int(*)( <b>AMP_STREAM_HDLR_s*</b> )	<b>Close</b>	The interface to close a stream handler
int(*)( <b>AMP_STREAM_HDLR_s*</b> , <b>UINT32</b> , <b>UINT8*</b> )	<b>Read</b>	The interface to read data from the stream
int(*)( <b>AMP_STREAM_HDLR_s*</b> , <b>UINT32</b> , <b>UINT8*</b> )	<b>Write</b>	The interface to write data to the stream
INT(*)( <b>AMP_STREAM_HDLR_s*</b> , <b>UINT64</b> , int)	<b>Seek</b>	The interface to seek the stream
INT64(*)( <b>AMP_STREAM_HDLR_s*</b> )	<b>GetPos</b>	The interface to retrieve the current stream position

Type	Field	Description
INT64 (*)(AMP_STREAM_HDLR_s*)	<b>GetLength</b>	The interface to retrieve the stream length
UINT64 (*)(AMP_STREAM_HDLR_s*)	<b>GetFreeSpace</b>	The interface to retrieve the available free space
int (*)(AMP_STREAM_HDLR_s*, UINT32, UINT32)	<b>Func</b>	The interface to execute special stream commands

Table 8-6. Definition of **AMP\_STREAM\_s** for Stream SDK6 API **AmpFileStream\_Create()**.

Confidential  
For HAOTEK Only

## 8.2.2 AmpFileStream\_Delete

### API Syntax:

**AmpFileStream\_Delete** ( AMP\_STREAM\_HDLR\_s \* hdlr)

### Function Description:

- This function is used to delete a file stream handler.

### Parameters:

Type	Parameter	Description
AMP_STREAM_HDLR_s*	hdlr	The handler of a file stream being removed ( <b>AMP_STREAM_HDLR_s</b> is defined in <code>Stream.h</code> ). Please refer to <a href="#">Section 8.2.1.2</a> for more details.

Table 8-7. Parameters for Stream SDK6 API **AmpFileStream\_Delete()**.

### Returns:

Return	Description
0 (0 (AMP_OK))	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 8-8. Returns for Stream SDK6 API **AmpFileStream\_Delete()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 8.2.3 AmpFileStream\_GetDefaultCfg

#### API Syntax:

**AmpFileStream\_GetDefaultCfg** ( AMP\_FILE\_STREAM\_CFG\_s \* config)

#### Function Description:

- This function is used to get the default configuration of a file stream handler.

#### Parameters:

Type	Parameter	Description
AMP_FILE_STREAM_CFG_s*	<b>config</b>	The returned configuration ( <b>AMP_FILE_STREAM_CFG_s</b> is defined in <code>File.h</code> ) Please refer to <a href="#">Section 8.2.1.1</a> for more details.

Table 8-9. Parameters for Stream SDK6 API **AmpFileStream\_GetDefaultCfg()**.

#### Returns:

Return	Description
0 (AMP_OK)	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 8-10. Returns for Stream SDK6 API **AmpFileStream\_GetDefaultCfg()**.

#### Example:

Please refer to the Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 8.2.4 AmpFileStream\_GetInitDefaultCfg

### API Syntax:

**AmpFileStream\_GetInitDefaultCfg** ( AMP\_FILE\_STREAM\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to get the default configuration for initializing the File Stream module.

### Parameters:

Type	Parameter	Description
AMP_FILE_STREAM_INIT_CFG_s*	<b>config</b>	The returned configuration. ( <b>AMP_FILE_STREAM_INIT_CFG_s</b> is defined in <code>File.h</code> ) Please refer to <a href="#">Section 8.2.4.1</a> for more details.

Table 8-11. Parameters for Stream SDK6 API **AmpFileStream\_GetInitDefaultCfg()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 8-12. Returns for Stream SDK6 API **AmpFileStream\_GetInitDefaultCfg()**.

### 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 AmpFileStream\_GetInitDefaultCfg > AMP\_FILE\_STREAM\_INIT\_CFG\_s

Type	Field	Description
UINT8*	<b>Buffer</b>	The work buffer of the File Stream module
UINT32	<b>BufferSize</b>	The size of the work buffer
UINT8	<b>MaxHdlr</b>	The maximum number of handlers held in the File Stream module

Table 8-13. Definition of **AMP\_FILE\_STREAM\_INIT\_CFG\_s** for Stream SDK6 API **AmpFileStream\_GetInitDefaultCfg()**.

## 8.2.5 AmpFileStream\_GetRequiredBufferSize

### API Syntax:

**AmpFileStream\_GetRequiredBufferSize** (UINT8 maxHdlr)

### Function Description:

- This function is used to get the required buffer size for initializing the File Stream module.

### Parameters:

Type	Parameter	Description
UINT8	<b>maxHdlr</b>	The maximum number of handlers held in the File Stream module

Table 8-14. Parameters for Stream SDK6 API **AmpFileStream\_GetRequiredBufferSize()**.

### Returns:

Return	Description
The required buffer size	The required buffer size

Table 8-15. Returns for Stream SDK6 API **AmpFileStream\_GetRequiredBufferSize()**.

### Example:

Please refer to the Unit Test document.

### See Also:

None

## 8.2.6 AmpFileStream\_Init

### API Syntax:

**AmpFileStream\_Init** ( AMP\_FILE\_STREAM\_INIT\_CFG\_s \* config)

### Function Description:

- This function is used to initialize the File Stream module.

### Parameters:

Type	Parameter	Description
AMP_FILE_STREAM_INIT_CFG_s*	<b>config</b>	The configuration used to initialize the File Stream module. ( <b>AMP_FILE_STREAM_INIT_CFG_s</b> is defined in <code>File.h</code> ) Please refer to <a href="#">Section 8.2.4.1</a> for more details.

Table 8-16. Parameters for Stream SDK6 API **AmpFileStream\_Init()**.

### Returns:

Return	Description
0 (AMP_OK)	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 8-17. Returns for Stream SDK6 API **AmpFileStream\_Init()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

# 9 Utility

## 9.1 Utility: Overview

This chapter provides the different utilities for different applications. This chapter also provides the description of the memory pool management function and the tool to dump the DSP log.

## 9.2 Utility: List of APIs

- AmpMath\_GetGCD
- AmpMath\_GetLCM
- AmpPtrList\_AddHead
- AmpPtrList\_AddTail
- AmpPtrList\_Create
- AmpPtrList\_Delete
- AmpPtrList\_GetAt
- AmpPtrList\_GetCount
- AmpPtrList\_GetRequiredBufSize
- AmpPtrList\_RemoveAll
- AmpPtrList\_RemoveAt
- AmpSync\_GetVinEofSystemTime
- AmpSync\_GetVinSofSystemTime
- AmpSync\_GetVinVsyncEofSystemTime
- AmpSync\_Init
- AmpSync\_WaitDChanVoutInt
- AmpSync\_WaitFChanVoutInt
- AmpSync\_WaitVinEofInt
- AmpSync\_WaitVinSofInt
- AmpSync\_WaitVinVsyncEofInt
- AmpUtil\_GetAlignedPool

## 9.2.1 AmpMath\_GetGCD

### API Syntax:

**AmpMath\_GetGCD** (UINT64 u, UINT64 v)

### Function Description:

- This function computes the GCD of two numbers.

### Parameters:

Type	Parameter	Description
UINT64	<b>u</b>	The first number
UINT64	<b>v</b>	The second number

Table 9-1. Parameters for Utility SDK6 API **AmpMath\_GetGCD()**.

### Returns:

Return	Description
GCD	Returns the GCD of u and v.

Table 9-2. Returns for Utility SDK6 API **AmpMath\_GetGCD()**.

### Example:

Please refer to the Unit Test document.

### See Also:

None

## 9.2.2 AmpMath\_GetLCM

### API Syntax:

**AmpMath\_GetLCM** (UINT64 \* v, UINT32 count)

### Function Description:

- This function helps to compute the LCM of an array of numbers.

### Parameters:

Type	Parameter	Description
UINT64 *	<b>v</b>	The array of numbers
UINT32	<b>count</b>	The size of the array

Table 9-3. Parameters for Utility SDK6 API **AmpMath\_GetLCM()**.

### Returns:

Return	Description
LCM	Returns the LCM of the numbers in v.

Table 9-4. Returns for Utility SDK6 API **AmpMath\_GetLCM()**.

### Example:

Please refer to the Unit Test document.

### See Also:

None

## 9.2.3 AmpPtrList\_AddHead

### API Syntax:

**AmpPtrList\_AddHead** (AMP\_PTR\_LIST\_HDLR\_s \* hdlr, void \* ptr)

### Function Description:

- This function adds a pointer to the head of a pointer list.

### Parameters:

Type	Parameter	Description
AMP_PTR_LIST_HDLR_s*	<b>hdlr</b>	The pointer list ( <b>AMP_PTR_LIST_HDLR_s</b> is defined in <code>PtrList.h</code> ) Please refer to <a href="#">Section 9.2.3.1</a> below for more details.
void *	<b>ptr</b>	The pointer to be added.

Table 9-5. Parameters for Utility SDK6 API **AmpPtrList\_AddHead()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-6. Returns for Utility SDK6 API **AmpPtrList\_AddHead()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

### 9.2.3.1 AmpPtrList\_AddHead > AMP\_PTR\_LIST\_HDLR\_s

Type	Field	Description
UINT32	Count	The number of elements that the pointer list keeps
UINT32	Limit	The maximum number of elements that the pointer list can keep

Table 9-7. Definition of **AMP\_PTR\_LIST\_HDLR\_s** for Utility SDK6 API **AmpPtrList\_AddHead()**.

## 9.2.4 AmpPtrList\_AddTail

### API Syntax:

**AmpPtrList\_AddTail** (AMP\_PTR\_LIST\_HDLR\_s \* hdlr, void \* ptr)

### Function Description:

- This function adds a pointer to the tail of a pointer list.

### Parameters:

Type	Parameter	Description
AMP_PTR_LIST_HDLR_s*	<b>hdlr</b>	The pointer list ( <b>AMP_PTR_LIST_HDLR_s</b> is defined in <code>PtrList.h</code> ) Please refer to <a href="#">Section 9.2.3.1</a> for more details.
void*	<b>ptr</b>	The pointer to be added

Table 9-8. Parameters for Utility SDK6 API **AmpPtrList\_AddTail()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-9. Returns for Utility SDK6 API **AmpPtrList\_AddTail()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 9.2.5 AmpPtrList\_Create

### API Syntax:

**AmpPtrList\_Create** (void \* buffer, UINT32 size, AMP\_PTR\_LIST\_HDLR\_s \*\* hdlr)

### Function Description:

- This function creates a pointer list from a buffer.

### Parameters:

Type	Parameter	Description
void*	<b>buffer</b>	The buffer
UINT32	<b>size</b>	The buffer size
AMP_PTR_LIST_HDLR_s**	<b>hdlr</b>	The double pointer to get the resulted pointer list ( <b>AMP_PTR_LIST_HDLR_s</b> is defined in <code>PtrList.h</code> ) Please refer to <a href="#">Section 9.2.3.1</a> for more details.

Table 9-10. Parameters for Utility SDK6 API **AmpPtrList\_Create()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE. e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-11. Returns for Utility SDK6 API **AmpPtrList\_Create()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 9.2.6 AmpPtrList\_Delete

### API Syntax:

**AmpPtrList\_Delete** (AMP\_PTR\_LIST\_HDLR\_s \* hdlr)

### Function Description:

- This function deletes a pointer list.

### Parameters:

Type	Parameter	Description
AMP_PTR_LIST_HDLR_s*	hdlr	The pointer list ( <b>AMP_PTR_LIST_HDLR_s</b> is defined in <code>PtrList.h</code> ) Please refer to <a href="#">Section 9.2.3.1</a> for more details.

Table 9-12. Parameters for Utility SDK6 API **AmpPtrList\_Delete()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-13. Returns for Utility SDK6 API **AmpPtrList\_Delete()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 9.2.7 AmpPtrList\_GetAt

### API Syntax:

**AmpPtrList\_GetAt** ( AMP\_PTR\_LIST\_HDLR\_s \* hdlr, UINT32 index, void \*\* ptr)

### Function Description:

- This function gets the index-th element of a pointer list.

### Parameters:

Type	Parameter	Description
AMP_PTR_LIST_HDLR_s*	<b>hdlr</b>	The pointer list ( <b>AMP_PTR_LIST_HDLR_s</b> is defined in <code>PtrList.h</code> ) Please refer to <a href="#">Section 9.2.3.1</a> for more details.
UINT32	<b>index</b>	The index of the element.
void**	<b>ptr</b>	The double pointer to get the element.

Table 9-14. Parameters for Utility SDK6 API **AmpPtrList\_GetAt()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-15. Returns for Utility SDK6 API **AmpPtrList\_GetAt()**.

### Example

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 9.2.8 AmpPtrList\_GetCount

### API Syntax:

**AmpPtrList\_GetCount** (AMP\_PTR\_LIST\_HDLR\_s \* hdlr)

### Function Description:

- This function gets the element count of a pointer list.

### Parameters:

Type	Parameter	Description
AMP_PTR_LIST_HDLR_s*	hdlr	The pointer list ( <b>AMP_PTR_LIST_HDLR_s</b> is defined in <code>PtrList.h</code> ) Please refer to <a href="#">Section 9.2.3.1</a> for more details.

Table 9-16. Parameters for Utility SDK6 API **AmpPtrList\_GetCount()**.

### Returns:

Return	Description
Count	Returns the element count.

Table 9-17. Returns for Utility SDK6 API **AmpPtrList\_GetCount()**.

### Example:

Please refer to the Unit Test document.

### See Also:

None

## 9.2.9 AmpPtrList\_GetRequiredBufferSize

### API Syntax:

**AmpPtrList\_GetRequiredBufSize** (UINT32 maxElement)

### Function Description:

- This function gets the required buffer size to create a pointer list.

### Parameters:

Type	Parameter	Description
UINT32	<b>maxElement</b>	The maximum number of elements that the pointer list can keep

Table 9-18. Parameters for Utility SDK6 API **AmpPtrList\_GetRequiredBufferSize()**.

### Returns:

Return	Description
Size	Returns the required buffer size in bytes.

Table 9-19. Returns for Utility SDK6 API **AmpPtrList\_GetRequiredBufferSize()**.

### Example:

Please refer to the Unit Test document.

### See Also:

None

## 9.2.10 AmpPtrList\_RemoveAll

### API Syntax:

**AmpPtrList\_RemoveAll** (AMP\_PTR\_LIST\_HDLR\_s \* hdlr)

### Function Description:

- This function removes all the elements of a pointer list.

### Parameters:

Type	Parameter	Description
AMP_PTR_LIST_HDLR_s *	hdlr	The pointer list ( <b>AMP_PTR_LIST_HDLR_s</b> is defined in <code>PtrList.h</code> ). Please refer to <a href="#">Section 9.2.3.1</a> for more details.

Table 9-20. Parameters for Utility SDK6 API **AmpPtrList\_RemoveAll()**.

### Example:

Please refer to the Unit Test document.

### See Also:

None

## 9.2.11 AmpPtrList\_RemoveAt

### API Syntax:

**AmpPtrList\_RemoveAt** (AMP\_PTR\_LIST\_HDLR\_s \* hdlr, UINT32 index)

### Function Description:

- This function removes the index-th element of a pointer list.

### Parameters:

Type	Parameter	Description
AMP_PTR_LIST_HDLR_s*	<b>hdlr</b>	The pointer list ( <b>AMP_PTR_LIST_HDLR_s</b> is defined in <code>PtrList.h</code> ) Please refer to <a href="#">Section 9.2.3.1</a> for more details.
UINT32	<b>index</b>	The index of the element to be deleted.

Table 9-21. Parameters for Utility SDK6 API **AmpPtrList\_RemoveAt()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-22. Returns for Utility SDK6 API **AmpPtrList\_RemoveAt()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 9.2.12 AmpSync\_GetVinEofSystemTime

### API Syntax:

**AmpSync\_GetVinEofSystemTime** (UINT32 channel, UINT32 \*time)

### Function Description:

- This function is used to get last Vin EOF system time.

### Parameters:

Type	Parameter	Description
UINT32	<b>channel</b>	channel
UINT32*	<b>time</b>	time

Table 9-23. Parameters for Utility SDK6 API **AmpSync\_GetVinEofSystemTime()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-24. Returns for Utility SDK6 API **AmpSync\_GetVinEofSystemTime()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



### 9.2.13 AmpSync\_GetVinSofSystemTime

#### API Syntax:

**AmpSync\_GetVinSofSystemTime** (UINT32 channel, UINT32\* time)

#### Function Description:

- This function is used to get last Vin SOF System time.

#### Parameters:

Type	Parameter	Description
UINT32	<b>channel</b>	channel
UINT32*	<b>time</b>	time

Table 9-25. Parameters for Utility SDK6 API **AmpSync\_GetVinSofSystemTime()**.

#### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-26. Returns for Utility SDK6 API **AmpSync\_GetVinSofSystemTime()**.

#### Example:

Please refer to the Unit Test document.

#### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 9.2.14 AmpSync\_GetVinVsyncEofSystemTime

### API Syntax:

**AmpSync\_GetVinVsyncEofSystemTime** (UINT32 channel, UINT32 \*time)

### Function Description:

- This function is used to get last Vin Vsync EOF system time.

### Parameters:

Type	Parameter	Description
UINT32	<b>channel</b>	Channel
UINT32*	<b>time</b>	time

Table 9-27. Parameters for Utility SDK6 API **AmpSync\_GetVinVsyncEofSystemTime()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-28. Returns for Utility SDK6 API **AmpSync\_GetVinVsyncEofSystemTime()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 9.2.15 AmpSync\_Init

### API Syntax:

**AmpSync\_Init** (UINT8 resource, void \*cfg)

### Function Description:

- This function is used to initialize the sync service.

### Parameters:

Type	Parameter	Description
UINT8	<b>resource</b>	Resource to sync
void*	<b>cfg</b>	Parameters

Table 9-29. Parameters for Utility SDK6 API **AmpSync\_Init()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-30. Returns for Utility SDK6 API **AmpSync\_Init()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 9.2.16 AmpSync\_WaitDChanVoutInt

### API Syntax:

**AmpSync\_WaitDChanVoutInt** (UINT32 intCount, UINT32 timeout)

### Function Description:

- This function waits for the digital channel VOUT syncs.

### Parameters:

Type	Parameter	Description
UINT32	<b>intCount</b>	Number of interrupts
UINT32	<b>timeout</b>	Timeout period for each waiting period

Table 9-31. Parameters for Utility SDK6 API **AmpSync\_WaitDChanVoutInt()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-32. Returns for Utility SDK6 API **AmpSync\_WaitDChanVoutInt()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 9.2.17 AmpSync\_WaitFChanVoutInt

### API Syntax:

**AmpSync\_WaitFChanVoutInt** (UINT32 intCount, UINT32 timeout)

### Function Description:

- This function waits for full function channel VOUT syncs.

### Parameters:

Type	Parameter	Description
UINT32	<b>intCount</b>	Number of interrupts
UINT32	<b>timeout</b>	Timeout period for each waiting

Table 9-33. Parameters for Utility SDK6 API **AmpSync\_WaitFChanVoutInt()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-34. Returns for Utility SDK6 API **AmpSync\_WaitFChanVoutInt()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 9.2.18 AmpSync\_WaitVinEofInt

### API Syntax:

**AmpSync\_WaitVinEofInt** (UINT32 channel, UINT32 intCount, UINT32 timeout)

### Function Description:

- This function waits for VIN EOF (last pixel) sync.

### Parameters:

Type	Parameter	Description
UINT32	<b>channel</b>	Channel number
UINT32	<b>intCount</b>	Number of interrupts
UINT32	<b>timeout</b>	Timeout period for each waiting

Table 9-35. Parameters for Utility SDK6 API **AmpSync\_WaitVinEofInt()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-36. Returns for Utility SDK6 API **AmpSync\_WaitVinEofInt()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 9.2.19 AmpSync\_WaitVinSofInt

### API Syntax:

**AmpSync\_WaitVinSofInt** (UINT32 channel, UINT32 intCount, UINT32 timeout)

### Function Description:

- This function waits for VIN SOF sync.

### Parameters:

Type	Parameter	Description
UINT32	<b>channel</b>	Channel number
UINT32	<b>intCount</b>	Number of interrupts
UINT32	<b>timeout</b>	Timeout period for each waiting

Table 9-37. Parameters for Utility SDK6 API **AmpSync\_WaitVinSofInt()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-38. Returns for Utility SDK6 API **AmpSync\_WaitVinSofInt()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

## 9.2.20 AmpSync\_WaitVinVsyncEofInt

### API Syntax:

**AmpSync\_WaitVinVsyncEofInt** (UINT32 channel, UINT32 intCount, UINT32 timeout)

### Function Description:

- This function is used to wait for Wait Vin Vsync EOF syncs..

### Parameters:

Type	Parameter	Description
UINT32	<b>channel</b>	channel
UINT32	<b>intCount</b>	Number of interrupts
UINT32	<b>timeout</b>	Timeout period for each waiting

Table 9-39. Parameters for Utility SDK6 API **AmpSync\_WaitVinVsyncEofInt()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-40. Returns for Utility SDK6 API **AmpSync\_WaitVinVsyncEofInt()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.



## 9.2.21 AmpUtil\_GetAlignedPool

### API Syntax:

**AmpUtil\_GetAlignedPool** (AMBA\_KAL\_BYTE\_POOL\_t \* BytePool, void \*\* AlignedPool, void \*\* Pool, UINT32 Size, UINT32 Alignment)

### Function Description:

- This function gets the aligned byte pool.

### Parameters:

Type	Parameter	Description
AMBA_KAL_BYTE_POOL_t *	<b>BytePool</b>	Byte Pool
void **	<b>AlignedPool</b>	Pool address after alignment
void **	<b>Pool</b>	Pool address before alignment
UINT32	<b>Size</b>	Pool Size
UINT32	<b>Alignment</b>	Alignment

Table 9-41. Parameters for Utility SDK6 API **AmpUtil\_GetAlignedPool()**.

### Returns:

Return	Description
AMP_OK	Success
All Others	AMP_ER_CODE_e. Please refer to <a href="#">Chapter 10</a> for more details on error codes.

Table 9-42. Returns for Utility SDK6 API **AmpUtil\_GetAlignedPool()**.

### Example:

Please refer to the Unit Test document.

### See Also:

Please refer to [Chapter 10](#) for more details on error codes.

# 10 System Errors

## 10.1 System Errors: Overview

This chapter lists the possible return values (**AMP**) when errors are encountered.

## 10.2 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

### 10.2.1 AMP\_ERROR\_GENERAL\_ERROR

**Error Value:**

AMP\_ERROE\_GENERAL\_ERROR

**Error Description:**

- General error.

### 10.2.2 AMP\_ERROR\_INCORRECT\_PARAM\_STRUCTURE

**Error Value:**

AMP\_ERROR\_INCORRECT\_PARAM\_STRUCTURE

**Error Description:**

- Incorrect structure used.

### 10.2.3 AMP\_ERROR\_INCORRECT\_PARAM\_VALUE\_RANGE

**Error Value:**

AMP\_ERROR\_INCORRECT\_PARAM\_VALUE\_RANGE

**Error Description:**

- Incorrect value range.

### 10.2.4 AMP\_ERROR\_OUT\_OF\_MEMORY

**Error Value:**

AMP\_ERROR\_OUT\_OF\_MEMORY

**Error Description:**

- Out of memory.

## 10.2.5 AMP\_ERROR\_RESOURCE\_INVALID

**Error Value:**

AMP\_ERROR\_RESOURCE\_INVALID

**Error Description:**

- Resource for the operation.

## 10.2.6 AMP\_ERROR\_FIFO\_TYPE\_MISMATCH

**Error Value:**

AMP\_ERROR\_FIFO\_TYPE\_MISMATCH

**Error Description:**

- Incorrect FIFO type.

## 10.2.7 AMP\_ERROR\_FIFO\_LOCKED

**Error Value:**

AMP\_ERROR\_FIFO\_LOCKED

**Error Description:**

- Try to read/write a locked FIFO.

## 10.2.8 AMP\_ERROR\_FIFO\_EMPTY

**Error Value:**

AMP\_ERROR\_FIFO\_EMPTY

**Error Description:**

- No entry in the FIFO.

### 10.2.9 AMP\_ERROR\_FIFO\_FULL

**Error Value:**

AMP\_ERROR\_FIFO\_FULL

**Error Description:**

- FIFO full.

### 10.2.10 AMP\_ERROR\_ILLEGAL\_OPERATION

**Error Value:**

AMP\_ERROR\_ILLEGAL\_OPERATION

**Error Description:**

- Illegal operation.

### 10.2.11 AMP\_ERROR\_ILLEGAL\_CONTAINER\_SOURCE

**Error Value:**

AMP\_ERROR\_ILLEGAL\_CONTAINER\_SOURCE

**Error Description:**

- Illegal container source.

### 10.2.12 AMP\_ERROR\_IO\_ERROR

**Error Value:**

AMP\_ERROR\_IO\_ERROR

**Error Description:**

- Stream IO error.

### 10.2.13 AMP\_ERROR\_OUT\_OF\_STORAGE

**Error Value:**

AMP\_ERROR\_OUT\_OF\_STORAGE

**Error Description:**

- Storage media full.

#### 10.2.14 AMP\_ERROR\_OPERATION\_ABORTED

**Error Value:**

AMP\_ERROR\_IO\_OPERATION\_ABORTED

**Error Description:**

- Operation aborted after a user aborts it.

#### 10.2.15 AMP\_ERROR\_OBJ\_ALREADY\_EXISTS

**Error Value:**

AMP\_ERROR\_OBJ\_ALREADY\_EXISTS

**Error Description:**

- An object has already existed.

#### 10.2.16 AMP\_ERROR\_OBJ\_UNAVAILABLE

**Error Value:**

AMP\_ERROR\_OBJ\_UNAVAILABLE

**Error Description:**

- A request object is unavailable.

#### 10.2.17 AMP\_ERROR\_OBJ\_CREATION\_FAILED

**Error Value:**

AMP\_ERROR\_OBJ\_CREATION\_FAILED

**Error Description:**

- Failed to create an object.

# Appendix 11 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 12 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	2 February 2015	Updated Chapter 1, 2, 3 register parameters; Added Sections 1.2.23, 3.2.21, 4.2.1, 5.6.1, 5.6.2, 5.6.3, 5.6.4, 5.6.5, 5.6.6, 5.6.7, 5.6.8, 5.6.9, 5.6.10, 5.6.11, 5.6.12, 5.6.13, 5.7.8, 5.7.9, 5.7.10, 5.7.11, 5.7.12, 5.7.13; Deleted Sections 1.2.25, 2.2.21
1.2	12 February 2015	Added Sections 10.2.13, 10.2.14 Updated Sections 1.2.13, 1.2.23, 4.2.1, 4.2.2, 4.2.3, 4.2.6, 4.2.7, 4.2.8, 5.6.1, 5.6.2, 5.6.7, 5.6.8, 5.6.9, 5.6.10, 5.6.11, 5.6.13, 5.7.8, 5.7.9, 5.7.10, 5.7.14.1, 5.7.16, 5.7.17, 5.7.18, 7.2.15.4, 10.2.3 Deleted Section 5.6.13.1
1.3	23 June 2015	Added Sections 1.2.24, 1.2.29, 1.2.36, 5.4.8.2, 5.4.8.3, 5.7.1.2, 7.2.6.4, 9.2.12, 9.2.13, 9.2.14, 9.2.15, 9.2.20, 10.2.15, 10.2.16, 10.2.17 Updated Sections 1.2.16, 1.2.21.1, 2.2.5, 2.2.10, 2.2.13.1, 2.2.13.2, 3.2.6.1, 3.2.6.2, 3.2.7.2, 3.2.9.2, 3.2.15, 5.2.1.1, 5.2.2.2, 5.2.2.3, 5.2.2.8, 5.2.2.9, 5.3.1.3, 5.3.1.4, 5.4.7, 5.6.1, 5.7.14.1, 5.8.16.1, 6.2.6.3, 6.2.22

Table A12-1. Revision History.