User's Manual

CONFIDENTIAL

ADSP TDM Renderer/Capture Plugin RCG3AHPLN0201ZDO

User's Manual

RCG3AHPLN0201ZDOE

Rev. 2.00 Dec, 2018

Renesas Electronics
www.renesas.com

How to Use This Manual

1. Purpose and Target Reader

This manual is designed to provide the user with an understanding of the interface specifications of the Software product. It is intended for users designing application systems incorporating the Software product. Please refer to the related documents with this product.

Use this Software after carefully reading the precautions. The precautions are stated in the main text of each section, at the end of each section, and in the usage precaution section.

The revision history summarizes major corrections and additions to the previous version. It does not cover all the changes. For details, refer to this manual.

2. Restrictions on the Use of this Software

This software is MIT license. The certificates from the licensor do not provide any assurances to users that the product performs reliably, intellectual property rights are protected, disputes are resolved by contract, and specifications are not subject to major changes. The user should use this software at his or her own risk.

3. Related Manuals

4. Technical Terms and Abbreviation

- Table of Contents -

- List of Figures -

- List of Tables -

1. Overview

This section provides an overview of the Time-Division Multiplexing (TDM) Renderer plugin. It contains TDM renderer and capture function.

1.1 Specifications Outline

TDM Renderer function plays the multiplexing audio signal based on the parameter that was set.

TDM Capture function capture/record the multiplexing audio signal based on the parameter that was set.

Table 1-1 Basic Specification

Item	Description
DSP	Cadence Design Systems, Inc. HiFi2
Compiler	Xtensa C and C++ Compiler (version 12.0.4)
Endian	Little Endian

Table 1-2 Supported TDM Renderer function Specifications

Item	Description			
Input data format	Channel number		PCM bit-width (fix-point)	
			16-bit	24-bit
	6ch	3 * 2ch	○	○
		1 * 6ch	○	○
	8ch	4 * 2ch	○	○
		1 * 8ch	○	○
Output data format	Time-division Multiplexing 16-bit/24-bit linear PCM (fixed point)			
Input Sampling frequency (Hz) supported	48000 / 44100 / 32000			
Output Sampling frequency (Hz) supported	48000 / 44100			
Number of channels supported	TDM format channel (6 / 8)			
Reentrant	Supported			
Other	-			
Restrictions	-			

Table 1-3 Support TDM Capture function Specification

Item	Description
Output data format	

Channel number		PCM bit-width (fix-point)	
		16-bit	24-bit
6ch	3 * 2ch	○	○
	1 * 6ch	○	○
8ch	4 * 2ch	○	○
	1 * 8ch	○	○

Input data format	Time-division Multiplexing 16-bit/24-bit linear PCM (fixed point)
Output Sampling frequency (Hz) supported	48000 / 44100 / 32000
Input Sampling frequency (Hz) supported	48000 / 44100
Number of channels supported	TDM format channel (6 / 8)
Reentrant	Supported
Other	-
Restrictions	-

Table 1-4 Memory Size Requirements

Memory type	Location	Memory area name		Size (in bytes)	
Instruction	ROM	Instruction area			53747
Data		Constant table area			
		Other area(Depended on the compiler)			
	RAM (TDM Capture)	Software work area			198284
		Area breakdown	Persistent area	Size breakdown	67208
			Scratch area		65536
			DTCM area		65536
			Built-in descriptor area		4
		User work area			34208
		Area breakdown	Output buffer	Size breakdown	32768
			Structure		1440
		Stack area			944
		Other area(Depended on the compiler)			0
	RAM (TDM Renderer)	Software work area			165516
		Area breakdown	Persistent area	Size breakdown	67208
			Scratch area		32768
			DTCM area		65536
			Built-in descriptor area		4
		User work area			34224
		Area breakdown	Input buffer	Size breakdown	32768
			Structure		1456
		Stack area			896
		Other area(Depended on the compiler)			0

[Note] Area whose location is shown as ROM in the location column can be included in RAM or ROM.
[Note] Area whose location is shown as RAM in the location column can be included in RAM only.
[Note] Built-in is a memory area to allocate descriptor memory, which need in the DMAC transfer type of plugin.

Table 1-5 Version Information

Item	Description
Library Version information	Version 1.0.0
API Version information	Version 1.0.0

1.2 Configuration

Figure 1-1 shows an example of the ADSP system configuration which uses renderer function.

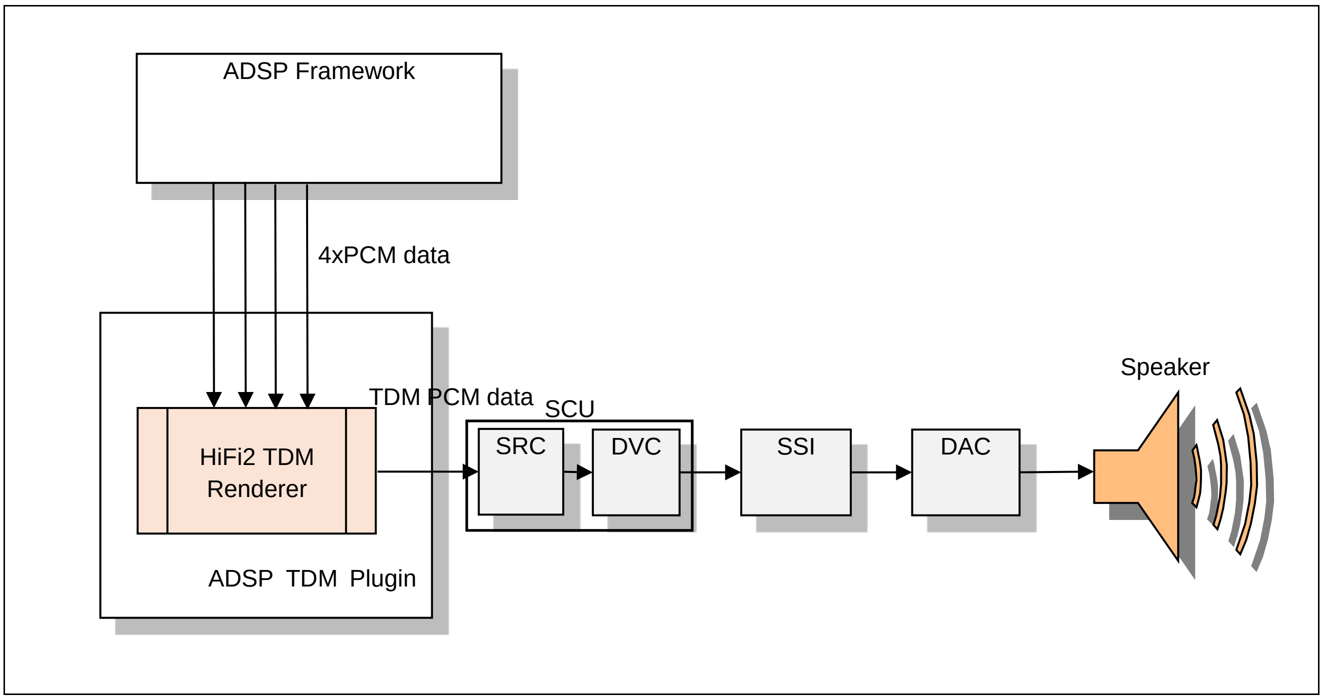


Figure 1-1 Example of the ADSP System Configuration for TDM renderer function

Figure 1-1 shows an example of the ADSP system configuration which uses capture function.

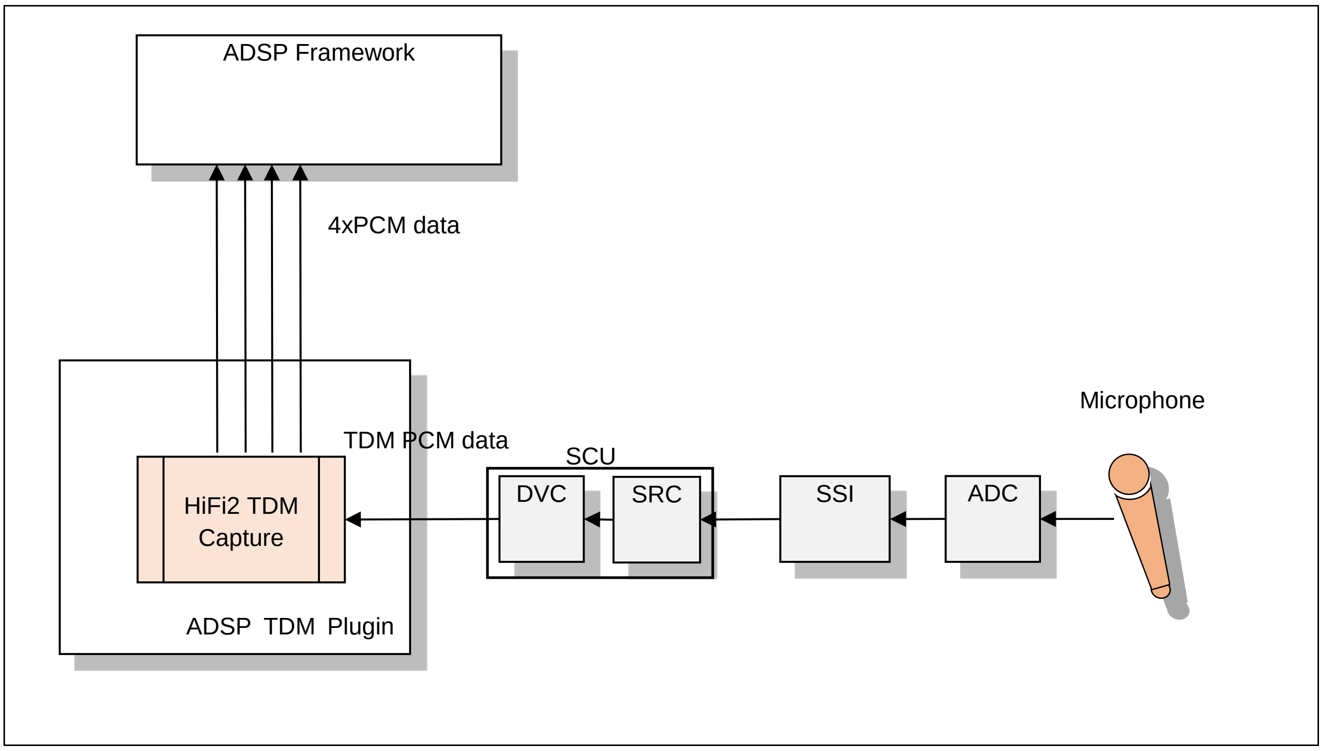


Figure 1-2 Example of the ADSP System Configuration for capture function

1. ADSP Framework

   It controls ADSP Plugin. It is software provided separately as Framework.

2. HiFi2 TDM Renderer (ADSP TDM Plugin)

   It performs merge multiple input PCM data and output to other audio device. It is this software set up as ADSP TDM Plugin.

3. HiFi2 TDM Capture (ADSP TDM Plugin)

   It performs split multiple output PCM data from TDM input received from other audio device. It is this software set up as ADSP TDM Plugin.

4. PCM data

   16-bit / 24-bit linear PCM data which is a processing by this software.

5. SCU

   It performs sampling rate converters (SRC) and volume control (DVC).

6. SSI (*)

   Send or receive audio data interfacing with a variety devices of offering I2C format.

7. DAC/ADC
   The DAC/ADC converts a digital 16-bit/24-bit linear PCM data into analog signal and vice versa.

2. Software Specifications

2.1 API specifications

A single interface function is used to access the plugin, with operation specified by command codes. Each library has a single C API call. The parameter definition for every library are same and is specified as below:

In TDM renderer case

Table 2-1 API Functions of TDM Renderer

xa_rel_tdm_rdr	
Description	This API is the only access function to the TDM renderer.
Syntax	XA_ERRORCODE xa_rel_tdm_rdr(xa_codec_handle_t p_xa_module_obj, WORD32 i_cmd, WORD32 i_idx, pVOID pv_value);
Parameters	p_xa_module_obj : Pointer to opaque API structure. i_cmd : Command. (defined in the supplied header files as) i_idx : Command subtype or index. (defined in the supplied header files as) pv_value : Pointer to the variable used to pass in, or get out properties, from state structure.
Returns	Error Code based on the success or failure of API command (defined in the supplied header files)

In TDM capture case

Table 2-2 API Functions of TDM Capture

xa_rel_tdm_cap	
Description	This API is the only access function to the capture.
Syntax	XA_ERRORCODE xa_rel_tdm_cap(xa_codec_handle_t p_xa_module_obj, WORD32 i_cmd, WORD32 i_idx, pVOID pv_value);
Parameters	p_xa_module_obj : Pointer to opaque API structure. i_cmd : Command. (defined in the supplied header files as) i_idx : Command subtype or index. (defined in the supplied header files as) pv_value : Pointer to the variable used to pass in, or get out properties, from state structure.
Returns	Error Code based on the success or failure of API command (defined in the supplied header files)

2.2 Command

Using API functions of the Table 2-1 and Table 2-2, it performs each processing by a combination of Command/Subcommand.

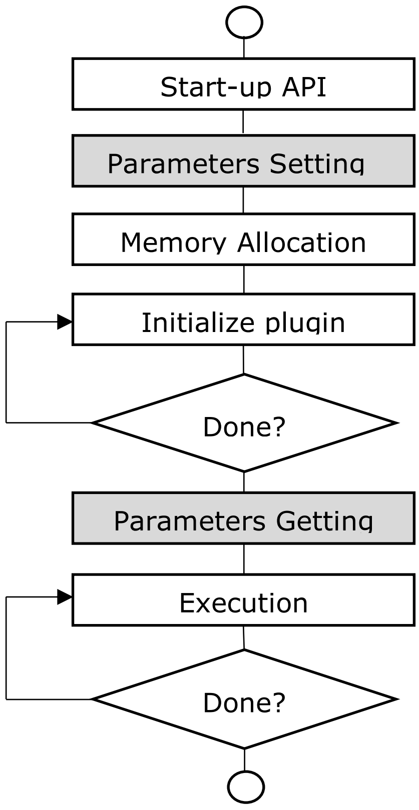


Figure 2-1 API command sequence overview

2.2.1 Command list

Below table presents commands used in renderer and capture case.

Table 2-3 List of supported none supported command, subcommand

Command	Sub command	R	C
XA_API_CMD_GET_LIB_ID_STRINGS	XA_CMD_TYPE_LIB_VERSION	○	○
	XA_CMD_TYPE_API_VERSION	○	○
XA_API_CMD_GET_API_SIZE	-	○	○
XA_API_CMD_INIT	XA_CMD_TYPE_INIT_API_PRE_CONFIG_PARAMS	○	○
	XA_CMD_TYPE_INIT_API_POST_CONFIG_PARAMS	○	○
	XA_CMD_TYPE_INIT_PROCESS	○	○
	XA_CMD_TYPE_INIT_DONE_QUERY	○	○
XA_API_CMD_SET_CONFIG_PARAM	XA_TDM_RDR_CONFIG_PARAM_PCM_WIDTH	○	–
	XA_TDM_RDR_CONFIG_PARAM_CHANNEL_MODE	○	–
	XA_TDM_RDR_CONFIG_PARAM_IN_SAMPLE_RATE	○	–
	XA_TDM_RDR_CONFIG_PARAM_FRAME_SIZE	○	–
	XA_TDM_RDR_CONFIG_PARAM_OUTPUT1	○	–
	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL1	○	–
	XA_TDM_RDR_CONFIG_PARAM_OUTPUT2	○	–
	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL2	○	–
	XA_TDM_RDR_CONFIG_PARAM_OUTPUT3*	○	–
	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL3*	○	–
	XA_TDM_RDR_CONFIG_PARAM_OUT_SAMPLE_RATE	○	–
	XA_TDM_RDR_CONFIG_PARAM_VOLUME_RATE	○	–
	XA_TDM_CAP_CONFIG_PARAM_PCM_WIDTH	–	○
	XA_TDM_CAP_CONFIG_PARAM_CHANNEL_MODE	–	○
	XA_TDM_CAP_CONFIG_PARAM_IN_SAMPLE_RATE	–	○
	XA_TDM_CAP_CONFIG_PARAM_FRAME_SIZE	–	○
	XA_TDM_CAP_CONFIG_PARAM_INPUT1	–	○
	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL1	–	○
	XA_TDM_CAP_CONFIG_PARAM_INPUT2	–	○
	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL2	–	○
	XA_TDM_CAP_CONFIG_PARAM_INPUT3*	–	○
	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL3*	–	○
	XA_TDM_CAP_CONFIG_PARAM_OUT_SAMPLE_RATE	–	○
	XA_TDM_CAP_CONFIG_PARAM_VOLUME_RATE	–	○
XA_API_CMD_GET_CONFIG_PARAM	XA_TDM_RDR_CONFIG_PARAM_PCM_WIDTH	○	–
	XA_TDM_RDR_CONFIG_PARAM_CHANNEL_MODE	○	–
	XA_TDM_RDR_CONFIG_PARAM_IN_SAMPLE_RATE	○	–
	XA_TDM_RDR_CONFIG_PARAM_FRAME_SIZE	○	–
	XA_TDM_RDR_CONFIG_PARAM_OUTPUT1	○	–
	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL1	○	–
	XA_TDM_RDR_CONFIG_PARAM_OUTPUT2	○	–
	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL2	○	–
	XA_TDM_RDR_CONFIG_PARAM_OUTPUT3*	○	–
	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL3*	○	–
	XA_TDM_RDR_CONFIG_PARAM_OUT_SAMPLE_RATE	○	–
	XA_TDM_RDR_CONFIG_PARAM_VOLUME_RATE	○	–
	XA_TDM_CAP_CONFIG_PARAM_PCM_WIDTH	–	○
	XA_TDM_CAP_CONFIG_PARAM_CHANNEL_MODE	–	○
	XA_TDM_CAP_CONFIG_PARAM_IN_SAMPLE_RATE	–	○
	XA_TDM_CAP_CONFIG_PARAM_FRAME_SIZE	–	○
	XA_TDM_CAP_CONFIG_PARAM_INPUT1	–	○
	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL1	–	○

	XA_TDM_CAP_CONFIG_PARAM_INPUT2	–	○
	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL2	–	○
	XA_TDM_CAP_CONFIG_PARAM_INPUT3*	–	○
	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL3*	–	○
	XA_TDM_CAP_CONFIG_PARAM_OUT_SAMPLE_RATE	–	○
	XA_TDM_CAP_CONFIG_PARAM_VOLUME_RATE	–	○
XA_API_CMD_GET_MEMTABS_SIZE	-	○	○
XA_API_CMD_SET_MEMTABS_PTR	-	○	○
XA_API_CMD_GET_N_MEMTABS	-	○	○
XA_API_CMD_GET_MEM_INFO_SIZE	-	○	○
XA_API_CMD_GET_MEM_INFO_ALIGNMENT	-	○	○
XA_API_CMD_GET_MEM_INFO_TYPE	-	○	○
XA_API_CMD_SET_MEM_PTR	-	○	○
XA_API_CMD_SET_INPUT_BYTES	-	○	○
XA_API_CMD_INPUT_OVER	-	○	○
XA_API_CMD_GET_CURIDX_INPUT_BUF	-	○	–
XA_API_CMD_EXECUTE	XA_CMD_TYPE_DO_EXECUTE	○	○
	XA_CMD_TYPE_DONE_QUERY	○	○
XA_API_CMD_GET_OUTPUT_BYTES	-	-	○

R: TDM Renderer C: TDM Capture

○ : Available

- : Omitted

* : Not applicable in current library version

2.2.1.1 Start-up API

Table 2-4 List of Initialization Commands

	upper stage : Command / lower step : Subcommand	Description
1	XA_API_CMD_GET_LIB_ID_STRINGS	Get the version of the library
	XA_CMD_TYPE_LIB_VERSION	
2	XA_API_CMD_GET_LIB_ID_STRINGS	Get the version of the API
	XA_CMD_TYPE_API_VERSION	
3	XA_API_CMD_GET_API_SIZE	Get the size of the API structure
	(NULL)	
4	XA_API_CMD_INIT	Set the default values of all the configuration parameters
	XA_CMD_TYPE_INIT_API_PRE_CONFIG_PARAMS	

2.2.1.2 Parameters setting

Table 2-5 List of Set Commands for renderer

	upper stage : Command / lower step : Subcommand	Description
1	XA_API_CMD_SET_CONFIG_PARAM	Set the input TDM PCM sample bit width to 16 or 24
	XA_TDM_RDR_CONFIG_PARAM_PCM_WIDTH	
2	XA_API_CMD_SET_CONFIG_PARAM	Set the input TDM PCM channel mode
	XA_TDM_RDR_CONFIG_PARAM_CHANNEL_MODE	
3	XA_API_CMD_SET_CONFIG_PARAM	Set the input TDM PCM sampling frequency (supported 32000/44100/48000 Hz)
	XA_TDM_RDR_CONFIG_PARAM_IN_SAMPLE_RATE	
4	XA_API_CMD_SET_CONFIG_PARAM	Set the input/output frame size
	XA_TDM_RDR_CONFIG_PARAM_FRAME_SIZE	
5	XA_API_CMD_SET_CONFIG_PARAM	Set the output destination Audio device 1st for TDM Renderer
	XA_TDM_RDR_CONFIG_PARAM_OUTPUT1	
6	XA_API_CMD_SET_CONFIG_PARAM	Set ADMA channel number usage for Audio device 1st (supported Audio-DMAC, Audio-DMAC-pp)
	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL1	
7	XA_API_CMD_SET_CONFIG_PARAM	Set the output destination Audio device 2nd for TDM Renderer
	XA_TDM_RDR_CONFIG_PARAM_OUTPUT2	
8	XA_API_CMD_SET_CONFIG_PARAM	Set ADMA channel number usage for Audio device 2nd (supported Audio-DMAC, Audio-DMAC-pp)
	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL2	
9	XA_API_CMD_SET_CONFIG_PARAM	Set the output PCM sampling frequency (supported 48000/44100 Hz)
	XA_TDM_RDR_CONFIG_PARAM_OUT_SAMPLE_RATE	
10	XA_API_CMD_SET_CONFIG_PARAM	Set the output PCM volume rate compare with input PCM (supported from 0 – 8 times)
	XA_TDM_RDR_CONFIG_PARAM_VOLUME_RATE	

Table 2-6 List of Set Commands for capture

	upper stage : Command / lower step : Subcommand	Description
1	XA_API_CMD_SET_CONFIG_PARAM XA_TDM_CAP_CONFIG_PARAM_PCM_WIDTH	Set the input TDM PCM sample bit width to 16 or 24
2	XA_API_CMD_SET_CONFIG_PARAM XA_TDM_CAP_CONFIG_PARAM_CHANNEL_MODE	Set the input TDM PCM channel mode
3	XA_API_CMD_SET_CONFIG_PARAM XA_TDM_CAP_CONFIG_PARAM_IN_SAMPLE_RATE	Set the input TDM PCM sampling frequency (supported 48000/44100 Hz)
4	XA_API_CMD_SET_CONFIG_PARAM XA_TDM_CAP_CONFIG_PARAM_FRAME_SIZE	Set the input/output frame size
5	XA_API_CMD_SET_CONFIG_PARAM XA_TDM_CAP_CONFIG_PARAM_INPUT1	Set the input source Audio device 1st for TDM Capture
6	XA_API_CMD_SET_CONFIG_PARAM XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL1	Set ADMA channel number usage for Audio device 1st (supported Audio-DMAC, Audio-DMAC-pp)
7	XA_API_CMD_SET_CONFIG_PARAM XA_TDM_CAP_CONFIG_PARAM_INPUT2	Set the input source Audio device 2nd for TDM Capture
8	XA_API_CMD_SET_CONFIG_PARAM XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL2	Set ADMA channel number usage for Audio device 2nd (supported Audio-DMAC, Audio-DMAC-pp)
9	XA_API_CMD_SET_CONFIG_PARAM XA_TDM_CAP_CONFIG_PARAM_OUT_SAMPLE_RATE	Set the output PCM sampling frequency (supported 32000/44100/48000 Hz)
10	XA_API_CMD_SET_CONFIG_PARAM XA_TDM_CAP_CONFIG_PARAM_VOLUME_RATE	Set the output PCM volume rate compare with input PCM (supported from 0 – 8 times)

2.2.1.3 Memory allocation

Table 2-7 List of Memory allocation Commands

	upper stage : Command / lower step : Subcommand	Description
1	XA_API_CMD_GET_MEMTABS_SIZE	Get the size of the memory structures to be allocated for the plugin tables
	(NULL)	
2	XA_API_CMD_SET_MEMTABS_PTR	Pass the memory structure pointer allocated for the tables
	(NULL)	
3	XA_API_CMD_INIT	Calculate the required sizes for all the memory blocks based on the setting specific parameters
	XA_CMD_TYPE_INIT_API_POST_CONFIG_PARAMS	
4	XA_API_CMD_GET_N_MEMTABS	Obtain the number of memory blocks required by plugin
	(NULL)	
5	XA_API_CMD_GET_MEM_INFO_SIZE	Get the size of the memory type being referred to by the index
	(NULL)	
6	XA_API_CMD_GET_MEM_INFO_ALIGNMENT	Get the alignment information of the memory type being referred to by the index
	(NULL)	
7	XA_API_CMD_GET_MEM_INFO_TYPE	Get the type of memory being referred to by the index
	(NULL)	
8	XA_API_CMD_SET_MEM_PTR	Set the pointer to the memory allocated for the referred index to the input value
	(NULL)	

2.2.1.4 Initialize plugin

Table 2-8 List of initialize commands

	upper stage : Command / lower step : Subcommand	Description
1	XA_API_CMD_SET_INPUT_BYTES	Set the number of bytes available in the input buffer
	(NULL)	
2	XA_API_CMD_INPUT_OVER	Signal to the plugin the end of the bit stream in renderer case
	(NULL)	
3	XA_API_CMD_INIT	Setup for the HW operation, and initialize state and configuration structure
	XA_CMD_TYPE_INIT_PROCESS	
4	XA_API_CMD_INIT	Check if the initialization process has completed
	XA_CMD_TYPE_INIT_DONE_QUERY	
5	XA_API_CMD_GET_CURIDX_INPUT_BUF	Get the number of input buffer bytes consumed
	(NULL)	

2.2.1.5 Parameters getting

Table 2-9 List of Get commands for renderer

upper stage : Command / lower step : Subcommand		Description
1	XA_API_CMD_GET_CONFIG_PARAM XA_TDM_RDR_CONFIG_PARAM_PCM_WIDTH	Get the input TDM PCM sample bit width
2	XA_API_CMD_GET_CONFIG_PARAM XA_TDM_RDR_CONFIG_PARAM_CHANNEL_MODE	Get the input TDM PCM channel mode
3	XA_API_CMD_GET_CONFIG_PARAM XA_TDM_RDR_CONFIG_PARAM_IN_SAMPLE_RATE	Get the input TDM PCM sampling frequency
4	XA_API_CMD_GET_CONFIG_PARAM XA_TDM_RDR_CONFIG_PARAM_FRAME_SIZE	Get the input/output frame size
5	XA_API_CMD_GET_CONFIG_PARAM XA_TDM_RDR_CONFIG_PARAM_OUTPUT1	Get TDM Renderer output destination Audio device 1st
6	XA_API_CMD_GET_CONFIG_PARAM XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL1	Get ADMA channel number usage for Audio device 1st
7	XA_API_CMD_GET_CONFIG_PARAM XA_TDM_RDR_CONFIG_PARAM_OUTPUT2	Get TDM Renderer output destination Audio device 2nd
8	XA_API_CMD_GET_CONFIG_PARAM XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL2	Get ADMA channel number usage for Audio device 2nd
9	XA_API_CMD_GET_CONFIG_PARAM XA_TDM_RDR_CONFIG_PARAM_OUT_SAMPLE_RATE	Get the output PCM sampling frequency
10	XA_API_CMD_GET_CONFIG_PARAM XA_TDM_RDR_CONFIG_PARAM_VOLUME_RATE	Get the output PCM volume rate compare with input PCM

Table 2-10 List of Get commands for capture

	upper stage : Command / lower step : Subcommand	Description
1	XA_API_CMD_GET_CONFIG_PARAM	Get the input TDM PCM sample bit width
	XA_TDM_CAP_CONFIG_PARAM_PCM_WIDTH	
2	XA_API_CMD_GET_CONFIG_PARAM	Get the input TDM PCM channel mode
	XA_TDM_CAP_CONFIG_PARAM_CHANNEL_MODE	
3	XA_API_CMD_GET_CONFIG_PARAM	Get the input TDM PCM sampling frequency
	XA_TDM_CAP_CONFIG_PARAM_IN_SAMPLE_RATE	
4	XA_API_CMD_GET_CONFIG_PARAM	Get the input/output frame size
	XA_TDM_CAP_CONFIG_PARAM_FRAME_SIZE	
5	XA_API_CMD_GET_CONFIG_PARAM	Get TDM Capture input source Audio device 1st
	XA_TDM_CAP_CONFIG_PARAM_INPUT1	
6	XA_API_CMD_GET_CONFIG_PARAM	Get ADMA channel number usage for Audio device 1st
	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL1	
7	XA_API_CMD_GET_CONFIG_PARAM	Get TDM Capture input destination Audio device 2nd
	XA_TDM_CAP_CONFIG_PARAM_INPUT2	
8	XA_API_CMD_GET_CONFIG_PARAM	Get ADMA channel number usage for Audio device 2nd
	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL2	
9	XA_API_CMD_GET_CONFIG_PARAM	Get the output PCM sampling frequency
	XA_TDM_CAP_CONFIG_PARAM_OUT_SAMPLE_RATE	
10	XA_API_CMD_GET_CONFIG_PARAM	Get the output PCM volume rate compare with input PCM
	XA_TDM_CAP_CONFIG_PARAM_VOLUME_RATE	

2.2.1.6 Execution

Table 2-11 List of execution commands

upper stage : Command / lower step : Subcommand		Description
1	XA_API_CMD_INPUT_OVER	Signal TDM Renderer/Capture the input data is over
	(NULL)	
2	XA_API_CMD_SET_INPUT_BYTES	Set the number of bytes available in the input buffer (only available in TDM Renderer)
	(NULL)	
3	XA_API_CMD_EXECUTE	Execute TDM Renderer/Capture plugin
	XA_CMD_TYPE_DO_EXECUTE	
4	XA_API_CMD_EXECUTE	Check if the execution process has completed
	XA_CMD_TYPE_DONE_QUERY	
5	XA_API_CMD_GET_OUTPUT_BYTES	Get the number of bytes output by the plugin in the last frame (only available in TDM Capture)
	(NULL)	
6	XA_API_CMD_GET_CURIDX_INPUT_BUF	Get the number of input buffer bytes consumed (only available in TDM Renderer)
	(NULL)	

2.2.2 Detail of Command Specifications

The next sections describe this library command functions by using the description format below.

Subcommand	Name of subcommand
Synopsis	Outlines the function.
Arguments	Describes the arguments for the function.
Restrictions	Provides information such as precautions in using the function.

[Note] This syntax format complies with ANSI-C.

2.2.2.1 XA_API_CMD_GET_LIB_ID_STRINGS command

Subcommand	XA_CMD_TYPE_LIB_VERSION	
Description	This command obtains the version of the library in the form of a string. The maximum length of the string that the library will provide is 30 bytes. Therefore the application shall pass a pointer to a buffer of a minimum size of 30 bytes. This command is optional	
Arguments	p_xa_module_obj	
	NULL	
	i_cmd	
	XA_API_CMD_GET_LIB_ID_STRINGS	
	i_idx	
	XA_CMD_TYPE_LIB_VERSION	
	pv_value	
	Pointer to a character buffer in which the version of the library is returned.	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	pv_value is NULL.
Restrictions	-	

Example:

```
char lib_version[30];
res = (*api_func)(NULL,
                  XA_API_CMD_GET_LIB_ID_STRINGS,
                  XA_CMD_TYPE_LIB_VERSION,
                  (pVOID) lib_version);
```

Subcommand	XA_CMD_TYPE_API_VERSION	
Description	This command obtains the version of the API in the form of a string. The maximum length of the string that the library will provide is 30 bytes. Therefore the application shall pass a pointer to a buffer of a minimum size of 30 bytes. This command is optional.	
Arguments	p_xa_module_obj	
	NULL	
	i_cmd	
	XA_API_CMD_GET_LIB_ID_STRINGS	
	i_idx	
	XA_CMD_TYPE_API_VERSION	
	pv_value	
	Pointer to a character buffer in which the version of the API is returned.	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	pv_value is NULL.
Restrictions	-	

Example:

```
char api_version[30];
res = (*api_func)(NULL,
                        XA_API_CMD_GET_LIB_ID_STRINGS,
                        XA_CMD_TYPE_API_VERSION,
                        (pVOID) api_version);
```

2.2.2.2 XA_API_CMD_GET_API_SIZE command

Subcommand	(None)	
Description	This command is used to obtain the size of the API structure, in order to allocate memory for the API structure.	
Arguments	p_xa_module_obj	
	NULL	
	i_cmd	
	XA_API_CMD_GET_API_SIZE	
	i_idx	
	NULL	
	pv_value	
	Pointer to API size variable.	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	pv_value is NULL.
Restrictions	The application shall allocate memory with an alignment of 4 bytes.	

Example:

```
WORD32 api_size;
res = (*api_func)(api_obj,
                  XA_CMD_TYPE_API_SIZE,
                  0,
                  &api_size);
```

2.2.2.3 XA_API_CMD_INIT command

Subcommand	XA_CMD_TYPE_INIT_API_PRE_CONFIG_PARAMS	
Description	This command is used to set the default value of the configuration parameters.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_INIT	
	i_idx	
	XA_CMD_TYPE_INIT_API_PRE_CONFIG_PARAMS	
	pv_value	
	NULL	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
Restrictions	-	

Example:

```
res = (*api_func)(api_obj,
                  XA_API_CMD_INIT,
                  XA_CMD_TYPE_INIT_API_PRE_CONFIG_PARAMS,
                  NULL);
```

Subcommand	XA_CMD_TYPE_INIT_API_POST_CONFIG_PARAMS	
Description	This command is used to calculate the sizes of all the memory blocks required by the application. It should occur after the plugin specific parameters have been set. If there are any parameters cannot be applied. Plugin returns a fatal error, or performs the change of these parameters automatically based on defined cases (i.e. enable SRC module if input sample rate sets to 32000 Hz, ...)	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_INIT	
	i_idx	
	XA_CMD_TYPE_INIT_API_POST_CONFIG_PARAMS	
	pv_value	
	NULL	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE (in TDM Capture) Or XA_TDM_RDR_CONFIG_FATAL_STATE (in TDM Renderer)	Incorrect sequence call (i.e. call before pre-configuration step or call before set memory table step)
	XA_TDM_CAP_EXEC_FATAL_INTERNAL (in TDM Capture) XA_TDM_RDR_EXEC_FATAL_INTERNAL (in TDM Renderer)	Invalid connection device setting path (i.e. setting SRC module for both device1 and device2), or lack of memory resource.
Restrictions	-	

Example:

```
res = (*api_func)(api_obj,
                  XA_API_CMD_INIT,
                  XA_CMD_TYPE_INIT_API_POST_CONFIG_PARAMS,
                  NULL);
```

Subcommand	XA_CMD_TYPE_INIT_PROCESS	
Description	Setup and start HW operation, and initialize state and configuration structure. No output data is created during initialization. In this state, plugin will check all hardware modules. If a module is busy, plugin will try to establish connection with next available one. If all module are busy, plugin will return error code.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_INIT	
	i_idx	
	XA_CMD_TYPE_INIT_PROCESS	
	pv_value	
	NULL	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_EXEC_FATAL_STATE (in TDM Capture) (XA_TDM_RDR_EXEC_FATAL_STATE (in TDM Renderer)	Incorrect sequence call (i.e. call before post-configuration step or without persistent/scratch buffer allocation, or without DTCM/Built-in descriptor memory allocation (in case of DMAC used)).
	XA_TDM_CAP_EXEC_FATAL_INTERNAL (in TDM Capture) XA_TDM_RDR_EXEC_FATAL_INTERNAL (in TDM Renderer)	Plugin has some abnormal cases happened from hardware modules (i.e. all hardware resource is busy).
Restrictions	-	

Example:

```
res = (*api_func)(api_obj,
                  XA_API_CMD_INIT,
                  XA_CMD_TYPE_INIT_PROCESS,
                  NULL);
```

Subcommand	XA_CMD_TYPE_INIT_DONE_QUERY	
Description	This command checks to see if the initialization process has completed. If it has, the flag value is set to one; else, it is set to zero. A pointer to the flag variable is passed as an argument.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_INIT	
	i_idx	
	XA_CMD_TYPE_INIT_DONE_QUERY	
	pv_value	
	Pointer to flag that indicates the completion of initialization process	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj or pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_EXEC_FATAL_STATE (in TDM Capture) (XA_TDM_RDR_EXEC_FATAL_STATE (in TDM Renderer)	Incorrect sequence call (i.e. call before post-configuration step)
Restrictions	-	

Example:

```
WORD32 done;
res = (*api_func)(api_obj,
                  XA_API_CMD_INIT,
                  XA_CMD_TYPE_INIT_DONE_QUERY,
                  &done);
```

2.2.2.4 XA_API_CMD_GET_MEMTABS_SIZE command

Subcommand	None	
Description	This command is used to obtain the size of the table used to hold the memory blocks required for the plugin operation. The API returns the total size of the required table. A pointer to the size variable is sent with this API command and the plugin writes the value to the variable.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_MEMTABS_SIZE	
	i_idx	
	NULL	
	pv_value	
	Pointer to memory size variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj or pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE (in TDM Capture) Or XA_TDM_RDR_CONFIG_FATAL_STATE (in TDM Renderer)	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example:

```
WORD32 memtab_size;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_MEMTABS_SIZE,
                  0,
                  &memtab_size);
```

2.2.2.5 XA_API_CMD_SET_MEMTABS_PTR command

Subcommand	None	
Description	This command is used to set the memory structure pointer in the library to the allocated value.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_MEMTABS_PTR	
	i_idx	
	NULL	
	pv_value	
	Allocated pointer	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj or pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj or pv_value is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE (in TDM Capture) Or XA_TDM_RDR_CONFIG_FATAL_STATE (in TDM Renderer)	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example:
```
pVOID memtab_ptr;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_MEMTABS_PTR,
                  0,
                  memtab_ptr);
```

2.2.2.6 XA_API_CMD_GET_N_MEMTABS command

Subcommand	None	
Description	This command is used to obtain the number of memory blocks needed by the plugin. This value is used as the iteration counter for the allocation of the memory blocks. A pointer to each memory block will be placed in the previously allocated memory tables. The pointer to the variable is passed to the API and the plugin writes the value to this variable.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_N_MEMTABS	
	i_idx	
	NULL	
	pv_value	
	Pointer to variable of number of memory blocks required to be allocated	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj or pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE (in TDM Capture) Or XA_TDM_RDR_CONFIG_FATAL_STATE (in TDM Renderer)	Incorrect sequence call (i.e. call before post-configuration step)
Restrictions	pv_value will be changed depend on channel mode and DMAC transfer type (using ADMAC or DMACPP)	

Example:

```
WORD32 n_memtab;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_N_MEMTABS,
                  0,
                  &n_memtab);
```

2.2.2.7 XA_API_CMD_GET_MEM_INFO_SIZE command

Subcommand	Memory index	
Description	This command obtains the size of the memory type being referred to by the index. The size in bytes is returned in the variable pointed to by the final argument.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_MEM_INFO_SIZE	
	i_idx	
	Index of the memory 0 - 1st Input Buffer (TDM Renderer) / 1st Output Buffer (TDM Capture) 1 - 2nd Input Buffer (TDM Renderer) / 2nd Output Buffer (TDM Capture) 2 - 3rd Input Buffer (TDM Renderer) / 3rd Output Buffer (TDM Capture) 3 - 4th Input Buffer (TDM Renderer) / 4th Output Buffer (TDM Capture) 4 - Persistent Area 5 - Scratch Area 6 - DTMC Area 7 - Built-in Area	
	pv_value	
	Pointer to memory size.	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj or pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned 4 bytes
	XA_TDM_CAP_CONFIG_FATAL_STATE (in TDM Capture) Or XA_TDM_RDR_CONFIG_FATAL_STATE (in TDM Renderer)	Incorrect sequence call (i.e. call before post-configuration step)
	XA_API_FATAL_INVALID_CMD_TYPE	Incorrect index
Restrictions	The index of DTCM and built-in area are only used in case of using ADMAC module to transfer data. And the index of input buffer will be affected by channel mode. So it may also affect to the other index memory.	

Example:

```
WORD32 mem_size;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_MEM_INFO_SIZE,
                  index,
                  &mem_size);
```

2.2.2.8 XA_API_CMD_GET_MEM_INFO_ALIGNMENT command

Subcommand	Memory index	
Description	This command gets the alignment information of the memory-type being referred to by the index. The alignment required in bytes is returned to the application.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_MEM_INFO_ALIGNMENT	
	i_idx	
	Index of the memory 0 - 1st Input Buffer (TDM Renderer) / 1st Output Buffer (TDM Capture) 1 - 2nd Input Buffer (TDM Renderer) / 2nd Output Buffer (TDM Capture) 2 - 3rd Input Buffer (TDM Renderer) / 3rd Output Buffer (TDM Capture) 3 - 4th Input Buffer (TDM Renderer) / 4th Output Buffer (TDM Capture) 4 - Persistent Area 5 - Scratch Area 6 - DTMC Area 7 - Built-in Area	
	pv_value	
	Pointer to the alignment info variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj or pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned 4 bytes
	XA_TDM_CAP_CONFIG_FATAL_STATE (in TDM Capture) Or XA_TDM_RDR_CONFIG_FATAL_STATE (in TDM Renderer)	Incorrect sequence call (i.e. call before post-configuration step)
	XA_API_FATAL_INVALID_CMD_TYPE	Incorrect index
Restrictions	The index of DTCM and built-in area are only used in case of using ADMAC module to transfer data. And the index of input buffer will be affected by channel mode. So it may also affect to the other index memory.	

Example:

```
WORD32 mem_align;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_MEM_INFO_ALIGNMENT,
                  index,
                  &mem_align);
```

2.2.2.9 XA_API_CMD_GET_MEM_INFO_TYPE command

Subcommand	Memory index	
Description	This command gets the alignment information of the memory-type being referred to by the index. The alignment required in bytes is returned to the application.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_MEM_INFO_TYPE	
	i_idx	
	Index of the memory 0 - 1st Input Buffer (TDM Renderer) / 1st Output Buffer (TDM Capture) 1 - 2nd Input Buffer (TDM Renderer) / 2nd Output Buffer (TDM Capture) 2 - 3rd Input Buffer (TDM Renderer) / 3rd Output Buffer (TDM Capture) 3 - 4th Input Buffer (TDM Renderer) / 4th Output Buffer (TDM Capture) 4 - Persistent Area 5 - Scratch Area 6 - DTMC Area 7 - Built-in Area	
	pv_value	
	Pointer to the memory type variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj or pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned 4 bytes
	XA_TDM_CAP_CONFIG_FATAL_STATE (in TDM Capture) Or XA_TDM_RDR_CONFIG_FATAL_STATE (in TDM Renderer)	Incorrect sequence call (i.e. call before post-configuration step)
	XA_API_FATAL_INVALID_CMD_TYPE	Incorrect index
Restrictions	The index of DTCM and built-in area are only used in case of using ADMAC module to transfer data. And the index of input buffer will be affected by channel mode. So it may also affect to the other index memory.	

Example:

```
WORD32 mem_type;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_MEM_INFO_TYPE,
                  index,
                  &mem_type);
```

2.2.2.10 XA_API_CMD_SET_MEM_PTR command

Subcommand	Memory index	
Description	This command passes to the plugin the pointer to the allocated memory. This is then stored in the memory tables structure allocated earlier. For the input and output buffers, it is legitimate to execute this command during the main plugin loop.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_MEM_PTR	
	i_idx	
	Index of the memory 0 - 1st Input Buffer (TDM Renderer) / 1st Output Buffer (TDM Capture) 1 - 2nd Input Buffer (TDM Renderer) / 2nd Output Buffer (TDM Capture) 2 - 3rd Input Buffer (TDM Renderer) / 3rd Output Buffer (TDM Capture) 3 - 4th Input Buffer (TDM Renderer) / 4th Output Buffer (TDM Capture) 4 - Persistent Area 5 - Scratch Area 6 - DTMC Area 7 - Built-in Area	
	pv_value	
	Pointer to the memory block	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj or pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes. pv_value is not aligned to required alignment for the requested memory block.
	XA_TDM_CAP_CONFIG_FATAL_STATE (in TDM Capture) Or XA_TDM_RDR_CONFIG_FATAL_STATE (in TDM Renderer)	Incorrect sequence call (i.e. call before post-configuration step)
	XA_API_FATAL_INVALID_CMD_TYPE	Incorrect index
Restrictions	The index of DTCM and built-in area are only used in case of using ADMAC module to transfer data. And the index of input buffer will be affected by channel mode. So it may also affect to the other index memory.	

Example:
```
pVOID addr;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_MEM_PTR,
                  index,
                  addr);
```

2.2.2.11 XA_API_CMD_INPUT_OVER command

Subcommand	None	
Description	This command is used to tell the plugin that the input signal is over. The execution or initialization step will continue in loop until it all the remaining input data is processed.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_INPUT_OVER	
	i_idx	
	NULL	
	pv_value	
	NULL	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj or pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_EXEC_FATAL_STATE (in TDM Capture) Or XA_TDM_RDR_EXEC_FATAL_STATE (in TDM Renderer)	Incorrect sequence call (i.e. call before initialization step – init process)
Restrictions	-	

Example:

```
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_INPUT_OVER,
                  0,
                  NULL);
```

2.2.2.12 XA_API_CMD_SET_INPUT_BYTES command

Subcommand	None	
Description	In TDM Capture this command will do nothing. The purpose of this command is filled the full list of standard API. In TDM Renderer this command will set number of bytes available in the input buffer.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_INPUT_BYTES	
	i_idx	
	The index of input buffer (only for TDM Renderer)	
	pv_value	
	Pointer to the input byte variable (Any value is OK with TDM Capture)	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj or pv_value is NULL
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes
	XA_TDM_RDR_EXEC_FATAL_STATE (only for TDM Renderer)	Input buffer is not ready, and have not init done
	XA_API_FATAL_INVALID_CMD_TYPE (only for TDM Renderer)	Incorrect index of input buffer
	XA_TDM_RDR_EXEC_FATAL_INPUT (only for TDM Renderer)	Invalid input buffer size (i.e. minus buffer size or buffer size is not align with sample size)
Restrictions	-	

Example:

```
WORD32 filled;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_INPUT_BYTES,
                  index,
                  &filled);
```

2.2.2.13 XA_API_CMD_GET_CURIDX_INPUT_BUF command

Subcommand	None	
Description	In TDM Capture, this command will return value 0 each time it’s called In TDM Renderer, this command will return number of input buffer bytes consumed	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CURIDX_INPUT_BUF	
	i_idx	
	The index of input buffer (only for TDM Renderer)	
	pv_value	
	Pointer to number variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj or pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_EXEC_FATAL_STATE (only for TDM Renderer)	Input buffer is not ready
	XA_API_FATAL_INVALID_CMD_TYPE (only for TDM Renderer)	Invalid index of input buffer
Restrictions	-	

Example:

```
WORD32 consumed;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CURIDX_INPUT_BUF,
                  index,
                  &consumed);
```

2.2.2.14 XA_API_CMD_EXECUTE command

Subcommand	XA_CMD_TYPE_DO_EXECUTE	
Description	This command execute the TDM Renderer/Capture plugin.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_EXECUTE	
	i_idx	
	XA_CMD_TYPE_DO_EXECUTE	
	pv_value	
	NULL	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_EXEC_FATAL_STATE (in TDM Capture) Or XA_TDM_RDR_EXEC_FATAL_STATE (in TDM Renderer)	Incorrect sequence call (i.e. call before initialization step) Or input / output buffer is not ready
	XA_TDM_CAP_EXEC_FATAL_INTERNAL (in TDM Capture) Or XA_TDM_RDR_EXEC_FATAL_INTERNAL (in TDM Renderer)	Hardware does not stop successfully
Restrictions	-	

Example:

```
res = (*api_func)(api_obj,
                  XA_API_CMD_EXECUTE,
                  XA_CMD_TYPE_DO_EXECUTE,
                  NULL);
```

Subcommand	XA_CMD_TYPE_DONE_QUERY	
Description	This command checks to see if the end of processing has been reached. If it is, the flag value is set to 1; else, it is set to zero. The pointer to the flag is passed as an argument. Processing by the plugin can continue for several invocations of the DO_EXECUTE command after the last input data has been passed to the plugin, so the application should not assume that the plugin has finished generating all its output until so indicated by this command.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_EXECUTE	
	i_idx	
	XA_CMD_TYPE_DONE_QUERY	
	pv_value	
	Pointer to the flag variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj or pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_EXEC_FATAL_STATE (in TDM Capture) Or XA_TDM_RDR_EXEC_FATAL_STATE (in TDM Renderer)	Incorrect sequence call (i.e. call before initialization step)
Restrictions	-	

Example:

```
WORD32 done;
res = (*api_func)(api_obj,
                  XA_API_CMD_EXECUTE,
                  XA_CMD_TYPE_DONE_QUERY,
                  &done);
```

2.2.2.15 XA_API_CMD_GET_OUTPUT_BYTES command

Subcommand	None	
Description	In TDM Renderer, this command will do nothing. The purpose of this command is fulfilled the standard APIs list. In TDM Capture, this command obtains the number of bytes output by the plugin during the last execution.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_OUTPUT_BYTES	
	i_idx	
	The index of output buffer (only for TDM Capture)	
	pv_value	
	Pointer to the flag variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj or pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_EXEC_FATAL_STATE (only for TDM Capture)	Incorrect sequence call (i.e. call before initialization step) Or output buffer is not ready
	XA_API_FATAL_INVALID_CMD_TYPE (only for TDM Capture)	Invalid index of output buffer
Restrictions	-	

Example

```
WORD32 produced;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_OUTPUT_BYTES,
                  index,
                  &produced);
```

2.2.2.16 XA_API_CMD_SET_CONFIG_PARAM command

2.2.2.16.1 Set configuration command for TDM Renderer

Subcommand	XA_TDM_RDR_CONFIG_PARAM_PCM_WIDTH	
Description	Set the TDM PCM sample bit width to 16 or 24 bits	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_PCM_WIDTH	
	pv_value	
	Pointer to the sample bit width variable (valid value: 16 or 24)	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_RDR_CONFIG_FATAL_PCM_WIDTH	TDM PCM sample bit width is invalid
Restrictions	-	

Example

```
WORD32 pcm_width;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_PCM_WIDTH,
                  &pcm_width);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_CHANNEL_MODE	
Description	Set TDM PCM channels mode	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_CHANNEL_MODE	
	pv_value	
	Pointer to the TDM channels mode variable XA_TDM_RDR_CHANNEL_MODE_2X4 : 4 stereo TDM data XA_TDM_RDR_CHANNEL_MODE_1X8 : 1 eight-channel TDM data XA_TDM_RDR_CHANNEL_MODE_2X3 : 3 stereo TDM data XA_TDM_RDR_CHANNEL_MODE_1X6 : 1 six-channel TDM data	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_RDR_CONFIG_FATAL_CHANNEL_MODE	Invalid TDM format
Restrictions	-	

Example:

```
WORD32 ch_mode;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_CHANNEL_MODE,
                  &ch_mode);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_IN_SAMPLE_RATE	
Description	Set input TDM PCM sampling frequency	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_IN_SAMPLE_RATE	
	pv_value	
	Pointer to the input sampling frequency variable (valid value: 32,000 / 44,100 / 48,000 Hz)	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_RDR_CONFIG_FATAL_SAMPLE_RATE	Input TDM PCM sampling frequency is out of range.
Restrictions	-	

Example

```
WORD32 sample_rate;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_IN_SAMPLE_RATE,
                  &sample_rate);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_FRAME_SIZE	
Description	Set input/output TDM PCM frame size in sample	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_FRAME_SIZE	
	pv_value	
	Pointer to frame size in sample variable (valid value: 512 / 1024 / 2048)	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_RDR_CONFIG_FATAL_FRAME_SIZE	TDM PCM frame size value is out of range.
Restrictions	-	

Example

```
WORD32 frame_size;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_FRAME_SIZE,
                  &frame_size);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_OUTPUT1	
Description	Set 1st output destination device for TDM Renderer.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_OUTPUT1	
	pv_value	
	Pointer to output destination value variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_RDR_CONFIG_FATAL_INVALID_OUTPUT	TDM PCM output device is out of range.
Restrictions	List of supported module:	

Macro	Value
SSI00	0
SSI10	10
SSI20	20
SSI30	30
SSI40	40
SSI90	90
SCU_SRCI0	110
SCU_SRCI1	111
SCU_SRCI3	113
SCU_SRCI4	114

Example:
```
WORD32 output_dev;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_OUTPUT1,
                  &output_dev);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL1	
Description	Set ADMA channel number usage for 1st Audio device.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL1	
	pv_value	
	Pointer to the Audio-DMAC / Audio-DMAC-peripheral-peripheral channels number ADMAC_CH[0-31] : Audio-DMAC usage ADMACPP_CH[0-28] : Audio-DMACpp usage	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_RDR_CONFIG_FATAL_DMACHANNEL	TDM PCM ADMA channel setting is out of range.
Restrictions	-	

Example:

```
WORD32 dma_channel;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL1,
                  &dma_channel);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_OUTPUT2	
Description	Set 2nd output destination device for TDM Renderer.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_OUTPUT2	
	pv_value	
	Pointer to output destination value variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_RDR_CONFIG_FATAL_INVALID_OUTPUT	TDM PCM output device is out of range.
Restrictions	List of supported module:	

Macro	Value
SSI00	0
SSI10	10
SSI20	20
SSI30	30
SSI40	40
SSI90	90
SCU_SRCI0	110
SCU_SRCI1	111
SCU_SRCI3	113
SCU_SRCI4	114

Example:

```
WORD32 output_dev;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_OUTPUT2,
                  &output_dev);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL2	
Description	Set ADMA channel number usage for 2nd Audio device.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL2	
	pv_value	
	Pointer to the Audio-DMAC / Audio-DMAC-peripheral-peripheral channels number ADMAC_CH[0-31] : Audio-DMAC usage ADMACPP_CH[0-28] : Audio-DMAC-pp usage	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_RDR_CONFIG_FATAL_DMACHANNEL	TDM PCM ADMA channel setting is out of range.
Restrictions	-	

Example:

```
WORD32 dma_channel;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL2,
                  &dma_channel);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_OUT_SAMPLE_RATE	
Description	Set output sample rate in Sampling Rate Converter (SRC) of Sampling Rate Converter Unit (SCU). If this setting is valid and non-zero value, SRC connection will be enabled even without setting connection device path. And the connection will automatically use the available Audio-DMAC channel. If this setting is zero, SRC module will not be used.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_OUT_SAMPLE_RATE	
	pv_value	
	Pointer to the output sampling frequency variable. Valid value: 0: disable SRC module 48,000/44,100 Hz: setting output sampling rate for SRC module	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_RDR_CONFIG_FATAL_SAMPLE_RATE	TDM PCM output sample rate is out of range.
Restrictions	-	

Example:
```
WORD32 sample_rate;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_OUT_SAMPLE_RATE,
                  &sample_rate);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_VOLUME_RATE	
Description	Set the output PCM volume rate in Digital Volume and Mute Function (DVC) of Sampling Rate Converter Unit (SCU). Any setting values except 0xFFFF FFFF (disable) will enabled DVC of SCU module and the connection will be established even without setting connection path.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_VOLUME_RATE	
	pv_value	
	Pointer to the volume ratio number (using Fix-point Q3.20): 0xFFFF FFFF : disable DVC module [0, 0x7F FFFF] : setting volume rate value	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_RDR_CONFIG_FATAL_VOLUME_RATE	TDM PCM volume rate value is out of range.
Restrictions	-	

Example:

```
WORD32 vol_rate;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_VOLUME_RATE,
                  &vol_rate);
```

2.2.2.16.2 Set configuration command for TDM Capture

Subcommand	XA_TDM_CAP_CONFIG_PARAM_PCM_WIDTH	
Description	Set TDM PCM sample bit width to 16 or 24 bits	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_PCM_WIDTH	
	pv_value	
	Pointer to the sample bit width variable (valid value: 16 or 24)	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_CAP_CONFIG_FATAL_PCM_WIDTH	TDM PCM sample width size is out of range.
Restrictions	-	

Example
```
WORD32 pcm_width;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_PCM_WIDTH,
                  &pcm_width);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_CHANNEL_MODE	
Description	Set TDM PCM channels mode	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_CHANNEL_MODE	
	pv_value	
	Pointer to the TDM channels mode variable XA_TDM_CAP_CHANNEL_MODE_2X4 : 4 stereo TDM data XA_TDM_CAP_CHANNEL_MODE_1X8 : 1 eight-channel TDM data XA_TDM_CAP_CHANNEL_MODE_2X3 : 3 stereo TDM data XA_TDM_CAP_CHANNEL_MODE_1X6 : 1 six-channel TDM data	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_CAP_CONFIG_FATAL_CHANNEL_MODE	Invalid TDM format
Restrictions	-	

Example

```
WORD32 ch;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_CHANNEL_MODE,
                  &ch);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_IN_SAMPLE_RATE	
Description	Set input sample rate in Sampling Rate Converter (SRC) of Sampling Rate Converter Unit (SCU). If this setting is valid and non-zero value, SRC connection will be enabled even without setting connection device path. And the connection will automatically use the available Audio-DMAC channel. If this setting is zero, SRC module will not be used.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_IN_SAMPLE_RATE	
	pv_value	
	Pointer to the input sampling frequency variable. Valid value: 0: disable SRC module 48,000/44,100 Hz: setting input sampling rate for SRC module	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_CAP_CONFIG_FATAL_SAMPLE_RATE	Input TDM PCM sampling frequency is out of range.
Restrictions	-	

Example

```
WORD32 sample_rate;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_IN_SAMPLE_RATE,
                  &sample_rate);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_FRAME_SIZE	
Description	Set input/output TDM PCM frame size in sample	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_FRAME_SIZE	
	pv_value	
	Pointer to frame size variable (valid value: 512 / 1024 / 2048)	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_CAP_CONFIG_FATAL_FRAME_SIZE	TDM PCM frame size value is out of range.
Restrictions	-	

Example

```
WORD32 frame_size;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_FRAME_SIZE,
                  &frame_size);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_INPUT1	
Description	Set 1st input source device for TDM Capture	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_INPUT1	
	pv_value	
	Pointer to the input device value variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_CAP_CONFIG_FATAL_INVALID_INPUT	TDM PCM input device is out of range.
Restrictions	List of supported module:	

Macro	Value
SSI00	0
SSI10	10
SSI20	20
SSI30	30
SSI40	40
SSI90	90
SCU_SRCI0	110
SCU_SRCI1	111
SCU_SRCI3	113
SCU_SRCI4	114

Example

```
WORD32 input_source;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_INPUT1,
                  &input_source);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL1	
Description	Set ADMA channel number usage for 1st Audio device.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL1	
	pv_value	
	Pointer to the Audio-DMAC / Audio-DMAC-peripheral-peripheral channels number ADMAC_CH[0-31] : Audio-DMAC usage ADMACPP_CH[0-28] : Audio-DMAC-pp usage	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_CAP_CONFIG_FATAL_DMACHANNEL	TDM PCM ADMA channel setting is out of range.
Restrictions	-	

Example

```
WORD32 dma_channel;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_CAP_CONFIG_PARAM_DMACHANNEL1,
                  &dma_channel);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_INPUT2	
Description	Set 2nd input source device for TDM Capture	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_INPUT2	
	pv_value	
	Pointer to the input device value variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_CAP_CONFIG_FATAL_INVALID_INPUT	TDM PCM input device is out of range.
Restrictions	List of supported module:	

Macro	Value
SSI00	0
SSI10	10
SSI20	20
SSI30	30
SSI40	40
SSI90	90
SCU_SRCI0	110
SCU_SRCI1	111
SCU_SRCI3	113
SCU_SRCI4	114

Example

```
WORD32 input_source;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_INPUT2,
                  &input_source);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL2	
Description	Set ADMA channel number usage for 2nd Audio device.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL2	
	pv_value	
	Pointer to the Audio-DMAC / Audio-DMAC-peripheral-peripheral channels number XA_TDM_CAP_ADMAC_CH[0-31] : Audio-DMAC usage XA_TDM_CAP_ADMACPP_CH[0-28] : Audio-DMAC-pp usage	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_CAP_CONFIG_NONFATAL_ERR_DMACHANNEL	TDM PCM ADMA channel setting is out of range.
Restrictions	-	

Example

```
WORD32 dma_channel;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_CAP_CONFIG_PARAM_DMACHANNEL2,
                  &dma_channel);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_OUT_SAMPLE_RATE	
Description	Set the PCM sampling frequency.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_OUT_SAMPLE_RATE	
	pv_value	
	Pointer to the output sampling frequency variable. Valid value: (32,000 / 44,100 / 48,000 Hz)	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_CAP_CONFIG_FATAL_SAMPLE_RATE	TDM PCM output sample rate is out of range.
Restrictions	-	

Example:

```
WORD32 sample_rate;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_OUT_SAMPLE_RATE,
                  &sample_rate);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_VOLUME_RATE	
Description	Set the output PCM volume rate in Digital Volume and Mute Function (DVC) of Sampling Rate Converter Unit (SCU). Any setting values except 0xFFFF FFFF (disable) will enabled DVC of SCU module and the connection will be established even without setting connection path.	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_SET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_VOLUME_RATE	
	pv_value	
	Pointer to the volume ratio number (using Fix-point Q3.20): 0xFFFF FFFF : disable DVC module [0, 0x7F FFFF] : setting volume rate value	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step or call after post-configuration step)
	XA_TDM_CAP_CONFIG_FATAL_VOLUME_RATE	TDM PCM volume rate value is out of range.
Restrictions	-	

Example:
```
WORD32 vol_rate;
res = (*api_func)(api_obj,
                  XA_API_CMD_SET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_VOLUME_RATE,
                  &vol_rate);
```

2.2.2.17 XA_API_CMD_GET_CONFIG_PARAM command

2.2.2.16.1 Get configuration command for TDM Renderer

Subcommand	XA_TDM_RDR_CONFIG_PARAM_PCM_WIDTH	
Description	Get the TDM PCM sample bit width setting	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_PCM_WIDTH	
	pv_value	
	Pointer to the sample bit width variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example

```
WORD32 pcm_width;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_PCM_WIDTH,
                  &pcm_width);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_CHANNEL_MODE	
Description	Get TDM PCM channels mode setting	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_CHANNEL_MODE	
	pv_value	
	Pointer to the TDM channels mode	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example:

```
WORD32 ch_mode;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_CHANNEL_MODE,
                  &ch_mode);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_IN_SAMPLE_RATE	
Description	Get input TDM PCM sampling frequency setting	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_IN_SAMPLE_RATE	
	pv_value	
	Pointer to the input sampling frequency variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example

```
WORD32 sample_rate;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_IN_SAMPLE_RATE,
                  &sample_rate);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_FRAME_SIZE	
Description	Get input/output TDM PCM frame size in sample setting	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_FRAME_SIZE	
	pv_value	
	Pointer to frame size in sample variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example

```
WORD32 frame_size;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_FRAME_SIZE,
                  &frame_size);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_OUTPUT1	
Description	Get 1st output destination device for TDM Renderer info	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_OUTPUT1	
	pv_value	
	Pointer to output destination value variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example:
```
WORD32 output_dev;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_OUTPUT1,
                  &output_dev);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL1	
Description	Get ADMA channel number usage for 1st Audio device info	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL1	
	pv_value	
	Pointer to the Audio-DMAC / Audio-DMAC-peripheral-peripheral channels variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example:
```
WORD32 dma_channel;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL1,
                  &dma_channel);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_OUTPUT2	
Description	Get 2nd output destination device for TDM Renderer info	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_OUTPUT2	
	pv_value	
	Pointer to output destination value variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example:

```
WORD32 output_dev;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_OUTPUT2,
                  &output_dev);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL2	
Description	Get ADMA channel number usage for 2nd Audio device info	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL2	
	pv_value	
	Pointer to the Audio-DMAC / Audio-DMAC-peripheral-peripheral channels variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example:

```
WORD32 dma_channel;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_DMACHANNEL2,
                  &dma_channel);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_OUT_SAMPLE_RATE	
Description	Get output sample rate setting	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_OUT_SAMPLE_RATE	
	pv_value	
	Pointer to the output sampling frequency variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example:

```
WORD32 sample_rate;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_OUT_SAMPLE_RATE,
                  &sample_rate);
```

Subcommand	XA_TDM_RDR_CONFIG_PARAM_VOLUME_RATE	
Description	Get the output PCM volume rate setting value	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_RDR_CONFIG_PARAM_VOLUME_RATE	
	pv_value	
	Pointer to the volume ratio number (using Fix-point Q3.20)	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_RDR_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example:

```
WORD32 vol_rate;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_RDR_CONFIG_PARAM_VOLUME_RATE,
                  &vol_rate);
```

2.2.2.16.2 Get configuration command for TDM Capture

Subcommand	XA_TDM_CAP_CONFIG_PARAM_PCM_WIDTH	
Description	Get TDM PCM sample bit width setting	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_PCM_WIDTH	
	pv_value	
	Pointer to the sample bit width variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example

```
WORD32 pcm_width;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_PCM_WIDTH,
                  &pcm_width);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_CHANNEL_MODE	
Description	Get TDM PCM channels mode setting	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_CHANNEL_MODE	
	pv_value	
	Pointer to the TDM channels mode variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example

```
WORD32 ch;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_CHANNEL_MODE,
                  &ch);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_IN_SAMPLE_RATE	
Description	Get the PCM sampling frequency setting value	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_IN_SAMPLE_RATE	
	pv_value	
	Pointer to the input sampling frequency variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example

```
WORD32 sample_rate;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_IN_SAMPLE_RATE,
                  &sample_rate);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_FRAME_SIZE	
Description	Get input/output TDM PCM frame size in sample setting	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_FRAME_SIZE	
	pv_value	
	Pointer to frame size variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example

```
WORD32 frame_size;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_FRAME_SIZE,
                  &frame_size);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_INPUT1	
Description	Get 1st input source device for TDM Capture info	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_INPUT1	
	pv_value	
	Pointer to the input destination value variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example
```
WORD32 input_source;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_INPUT1,
                  &input_source);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL1	
Description	Get ADMA channel number usage for 1st Audio device info	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL1	
	pv_value	
	Pointer to the Audio-DMAC / Audio-DMAC-peripheral-peripheral channels variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example

```
WORD32 dma_channel;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_CAP_CONFIG_PARAM_DMACHANNEL1,
                  &dma_channel);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_INPUT2	
Description	Get 2nd input source device for TDM Capture info	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_INPUT2	
	pv_value	
	Pointer to the input destination value variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example
```
WORD32 input_source;
res = (*api_func)(api_obj,
                   XA_API_CMD_GET_CONFIG_PARAM,
                   XA_TDM_CAP_CONFIG_PARAM_INPUT2,
                   &input_source);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL2	
Description	Get ADMA channel number usage for 2nd Audio device info	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_DMACHANNEL2	
	pv_value	
	Pointer to the Audio-DMAC / Audio-DMAC-peripheral-peripheral channels variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example

```
WORD32 dma_channel;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_CAP_CONFIG_PARAM_DMACHANNEL2,
                  &dma_channel);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_OUT_SAMPLE_RATE	
Description	Get output sample rate setting value	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_OUT_SAMPLE_RATE	
	pv_value	
	Pointer to the output sampling frequency variable	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example:

```
WORD32 sample_rate;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_OUT_SAMPLE_RATE,
                  &sample_rate);
```

Subcommand	XA_TDM_CAP_CONFIG_PARAM_VOLUME_RATE	
Description	Get the output PCM volume rate setting value	
Arguments	p_xa_module_obj	
	Pointer to API Structure.	
	i_cmd	
	XA_API_CMD_GET_CONFIG_PARAM	
	i_idx	
	XA_TDM_CAP_CONFIG_PARAM_VOLUME_RATE	
	pv_value	
	Pointer to the volume ratio number (using Fix-point Q3.20)	
Return value	XA_NO_ERROR	Normally ends.
	XA_API_FATAL_MEM_ALLOC	p_xa_module_obj / pv_value is NULL.
	XA_API_FATAL_MEM_ALIGN	p_xa_module_obj is not aligned to 4 bytes.
	XA_TDM_CAP_CONFIG_FATAL_STATE	Incorrect sequence call (i.e. call before pre-configuration step)
Restrictions	-	

Example:

```
WORD32 vol_rate;
res = (*api_func)(api_obj,
                  XA_API_CMD_GET_CONFIG_PARAM,
                  XA_TDM_CAP_CONFIG_PARAM_VOLUME_RATE,
                  &vol_rate);
```

2.3 Structures

Table 2-12 lists the structures for this software. The user should reserve areas required for these structures. For detailed specifications of these input structures, refer to Section 2.3.1.

Table 2-12 Structures

Structure name	Size	Outline
XARelTDMrdr	1456 bytes	API's structure to stores the information of API
XARelTDMcap	1440 bytes	API's structure to stores the information of API

2.3.1 XARelTDMrdr type structure

The XARelTDMrdr type structure is the work area used by the TDM Renderer of TDM plugin. When using this plugin, secure the area with the application program. It's not necessary to refer to this area because it only contains the internal variables and working buffers of the plugin. Make sure not to change the value of this area with the application program.

Table 2-13 XARelTDMrdr type structure information

Member name	Outline
pVOID pMem_tabs	Pointer to memory tables
WORD32 persist_size	Size of persistent memory
WORD32 descript_size	Descriptor memory size
WORD32 ring_size	Total size of ring buffer in sample
WORD32 sample_size	Size of PCM sample in byte (respect channels and PCM width)
WORD32 input_total	Number of input port based on channels mode of TDM plugin
WORD32 channels	Format channel of input PCM data
relTDMrdr_Parameters parameters	Parameter structure of TDM renderer plugin
DMAC_SETTING dma_params	ADMAC parameters structure
WORD32 output1_type	1st audio device type
WORD32 output2_type	2nd audio device type
WORD32 dma1_type	1st DMAC connection type
WORD32 dma2_type	2nd DMAC connection type
SSIU_SSI_MODULE ssi_module	SSI module information
SRC_MODULES src_module	SRC module information
CMD_MODULE cmd_module	CMD module information
Fifo_modules fifo_module	FIFO module information
WORD32 state	TDM renderer state
WORD32 dmac_stage	ADMAC stage flag
WORD32 hw_module	Store module information used in plugin
WORD32 write_idx	FIFO writing position
WORD32 read_idx	Software reading position
WORD32 filled	Number of sample present in the buffer
WORD32 merging_count	Number of bytes plugin has written into scratch area
WORD32 port_filled[4]	Number of bytes port has been submitted from user
WORD32 merging_done	Flag to tell plugin that merging process is done or not yet
WORD32 consumed[4]	Number of byte consumed in each port
XosEvent relrdr_event	TDM Renderer polling event
XosThread relrdr_thread	TDM Renderer polling thread

2.3.2 XARelTDMcap type structure

The XARelTDMcap type structure is the work area used by the TDM Capture of TDM plugin. When using this plugin, secure the area with the application program. It's not necessary to refer to this area because it only contains the internal variables and working buffers of the plugin. Make sure not to change the value of this area with the application program.

Table 2-14 XARelTDMcap type structure information

Member name	Outline
pVOID pMem_tabs	Pointer to memory tables
WORD32 persist_size	Size of persistent memory
WORD32 descript_size	Descriptor memory size
WORD32 ring_size	Total size of ring-buffer in sample
WORD32 sample_size	Size of PCM sample in byte (respect channels and PCM width)
WORD32 output_total	Number of output port based on channel mode of plugin
WORD32 channels	Format channel of input PCM data
relTDMcap_Parameters parameters	Parameter structure of TDM Capturer plugin
DMAC_SETTING dma_params	ADMAC parameters structure
WORD32 input1_type	1st audio device type
WORD32 input2_type	2nd audio device type
WORD32 dma1_type	1st DMAC connection type
WORD32 dma2_type	2nd DMAC connection type
SSIU_SSI_MODULE ssi_master	SSI module master information
SSIU_SSI_MODULE ssi_slave	SSI module slaver information
SRC_MODULES src_module	SRC module information
CMD_MODULE cmd_module	CMD module information
Fifo_modules fifo_module	FIFO module information
WORD32 state	TDM Capture plugin current state
WORD32 dmac_stage	ADMAC stage flag
WORD32 hw_module	Store module information used in plugin
WORD32 head_idx	Head index of ring buffer
WORD32 tail_idx	Tail index of ring buffer
WORD32 filled	Number of samples present in the buffer
WORD32 transfered_idx	Index of output port has been transferred in the last execution
WORD32 transferred[4]	Number of byte transferred by plugin for each port
XosEvent relcap_event	TDM Capture polling event
XosThread relcap_thread	TDM Capture polling thread

2.4 Memory Specifications

This section describes the memory areas used by this software.

2.4.1 Persistent Area

Table 2-15 Persistent Area Description

Item	Area which always holds values when this software is used. If the user manipulates this area after initialization, the correct execution of this software is not ensured.
Symbol name	- (freely defined by the user)
Size	Obtain the actually required size with 2.2.2.7
Area reservation	The user should reserve this area.
Allocation	This area is included in RAM.
Alignment	Align this area on a 4-byte boundary.

2.4.2 Stack Area

This software does not use a stack area.

2.4.3 Heap Area

This software does not use a heap area.

2.4.4 Input Buffer

Input buffer only is used in the TDM Renderer case.

Table 2-16 Input Buffer Description

Item	Area which stores inputs from this software. The input buffer contains 16-bit or 24-bit linear PCM data (hereinafter called PCM data). If the user manipulates this area during rendering processing, the normal execution of the program cannot be ensured.
Symbol name	- (freely defined by the user)
Size	Please secure more than size with 2.2.2.7 (a multiple of 2.2.2.7).
Area reservation	The user should reserve this area. The user can freely use this area after the rendering of one block.
Allocation	This area is included in RAM.
Alignment	Align this area on a 4-byte boundary.

2.4.5 Output Buffer

Output buffer only is used in the TDM Capture case.

Table 2-17 Output Buffer Description

Item	Area which stores outputs from this software. The output buffer contains 16-bit or 24-bit linear PCM data (hereinafter called PCM data). If the user manipulates this area during rendering processing, the normal execution of the program cannot be ensured.
Symbol name	- (freely defined by the user)
Size	Please secure more than size with 2.2.2.7 (a multiple of 2.2.2.7).
Area reservation	The user should reserve this area. The user can freely use this area after the rendering of one block.
Allocation	This area is included in RAM.
Alignment	Align this area on a 4-byte boundary.

(1) Input/ Output data storage method

Data is input/ output in the formats as shown in Figure 2-4(consecutive buffers are specified for the channels). The input/output buffer (memory) stores data in 2-byte (16-bit) units. The byte order for accessing the buffer is little endian (see Figure 2-2).

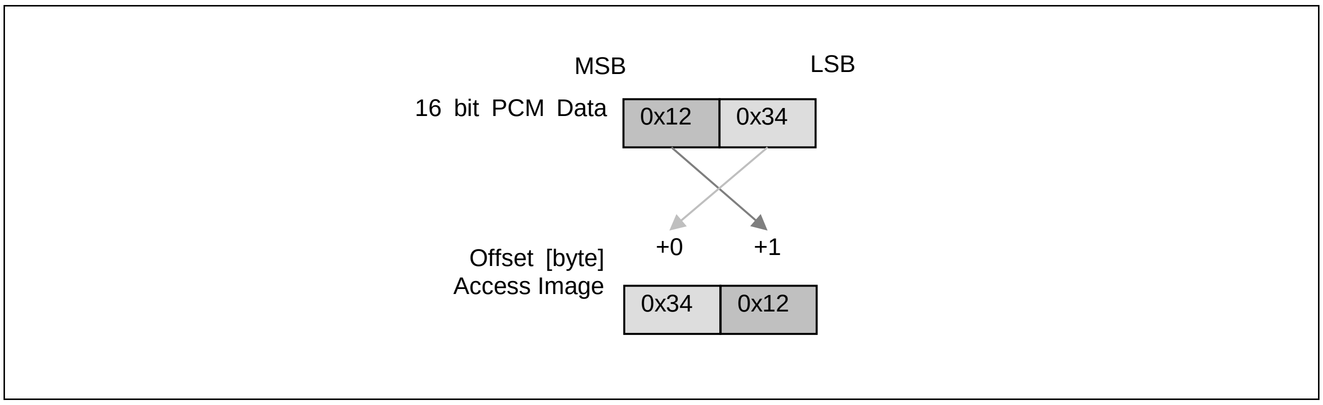


Figure 2-2 PCM 16-bit Data Access (Little Endian Mode)

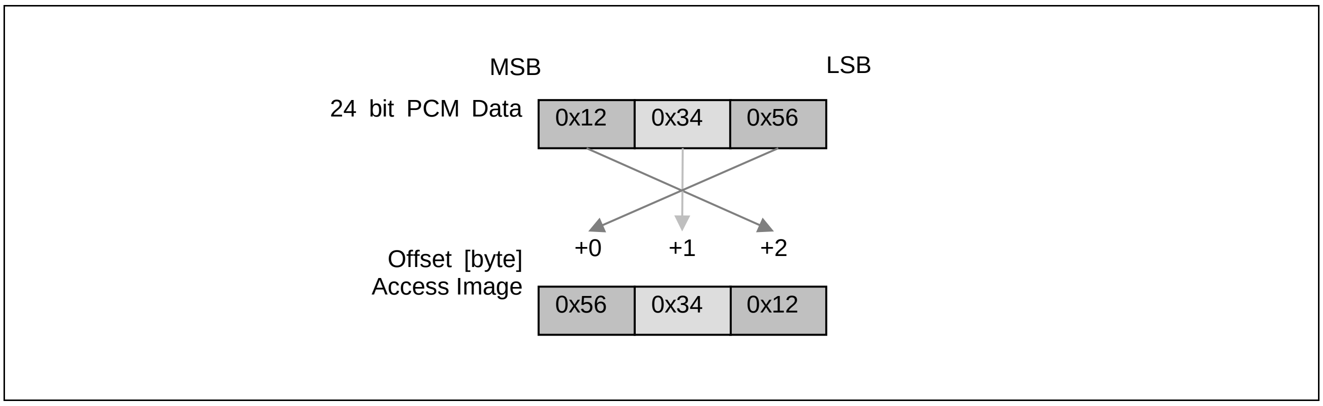
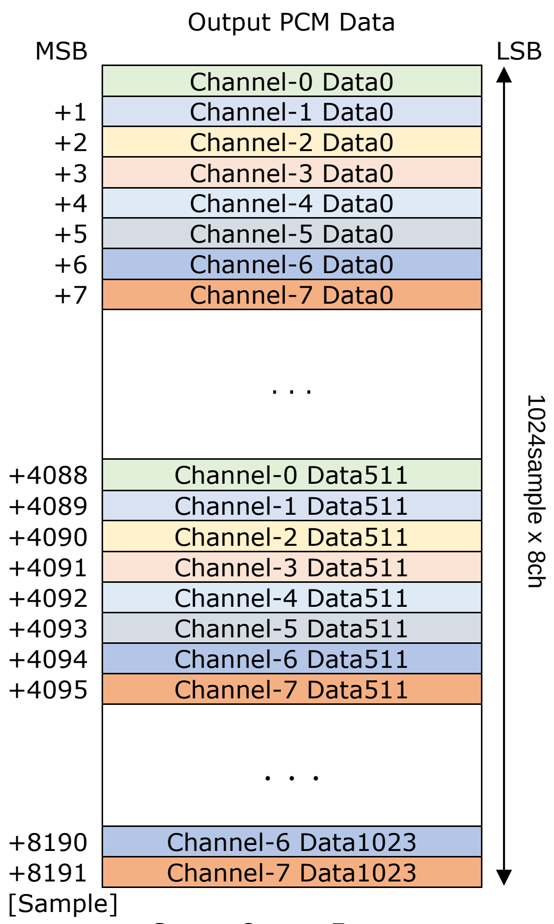


Figure 2-3 PCM 24-bit Data Access (Little Endian Mode)



Stereo Output Format

Figure 2-4 Output Formats

2.5 Error Processing

This software's functions return the error codes listed in Table 2-19.

2.5.1 Error codes

Below are the error codes for this software.

Table 2-18 Error Codes for TDM Renderer

Error code (32bit)	Value	Description
[1] XA_NO_ERROR	0x00000000	The processing results are normal. The process has terminated normally.
[2] XA_API_FATAL_MEM_ALLOC	0xFFFF8000	Abnormality has occurred, which disables process continuation. An address of API structure was specified at the argument is NULL, the program execution is incorrect. Because it becomes the common API error, please check the correct procedure.
[3] XA_API_FATAL_MEM_ALIGN	0xFFFF8001	Abnormality has occurred, which disables process continuation. An address of API structure was specified at the argument does not 4 byte align. Because it becomes the common API error, please check the correct procedure.
[4] XA_API_FATAL_INVALID_CMD	0xFFFF8002	Abnormality has occurred, which disables process continuation. The command was specified at the argument does not support. Because it becomes the common API error, please check the correct procedure.
[5] XA_API_FATAL_INVALID_CMD_TYPE	0xFFFF8003	Abnormality has occurred, which disables process continuation. The subcommand was specified at the argument does not support. Because it becomes the common API error, please check the correct procedure.
[6] XA_TDM_RDR_EXEC_FATAL_STATE	0xFFFF9080	Abnormality has occurred, which disables process continuation. The command does not follow procedure. Because it becomes the common API error, please check the correct procedure.
[7] XA_TDM_RDR_EXEC_FATAL_INPUT	0xFFFF9081	Abnormality has occurred, which disables process continuation. The input size is not align with sample size. Because it becomes the common API error, please check the correct size of input buffers.
[8] XA_TDM_RDR_EXEC_FATAL_INTERNAL	0xFFFF9082	Abnormality has occurred, which disables process continuation. Some of setting becomes incorrect after combination (out of memory, hardware modules are not available...). Because it becomes the common API error, please check the correct parameters and make sure the resource is validity.

[9] XA_TDM_RDR_CONFIG_FATAL_STATE	0xFFFF8880	Abnormality has occurred, which disables process continuation. The command does not follow procedure. Because it becomes the common API error, please check the correct procedure.
[10] XA_TDM_RDR_CONFIG_FATAL_PCM_WIDTH	0xFFFF8881	It is an error for TDM Renderer specifications out of the range. The pcm width value was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16)
[11] XA_TDM_RDR_CONFIG_FATAL_CHANNEL_MODE	0xFFFF8882	It is an error for TDM Renderer specifications out of the range. The channel mode value was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16)
[12] XA_TDM_RDR_CONFIG_FATAL_SAMPLE_RATE	0xFFFF8883	It is an error for TDM Renderer specifications out of the range. The sample rate value was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16)
[13] XA_TDM_RDR_CONFIG_FATAL_FRAME_SIZE	0xFFFF8884	It is an error for TDM Renderer specifications out of the range. The frame size was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16)
[14] XA_TDM_RDR_CONFIG_FATAL_INVALID_OUTPUT	0xFFFF8885	It is an error for TDM Renderer specifications out of the range. The output value was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16
[15] XA_TDM_RDR_CONFIG_FATAL_DMACHANNEL	0xFFFF8886	It is an error for TDM Renderer specifications out of the range. The adma channel value was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16)
[16] XA_TDM_RDR_CONFIG_FATAL_VOLUME_RATE	0xFFFF8887	It is an error for TDM Renderer specifications out of the range. The volume rate value was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16)
[17]	Others	Reserved

Table 2-19 Error Codes for TDM Capture

Error code (32bit)	Value	Description
[1] XA_NO_ERROR	0x00000000	The processing results are normal. The process has terminated normally.
[2] XA_API_FATAL_MEM_ALLOC	0xFFFF8000	Abnormality has occurred, which disables process continuation. An address of API structure was specified at the argument is NULL, the program execution is incorrect. Because it becomes the common API error, please check the correct procedure.
[3] XA_API_FATAL_MEM_ALIGN	0xFFFF8001	Abnormality has occurred, which disables process continuation. An address of API structure was specified at the argument does not 4 byte align. Because it becomes the common API error, please check the correct procedure.
[4] XA_API_FATAL_INVALID_CMD	0xFFFF8002	Abnormality has occurred, which disables process continuation. The command was specified at the argument does not support. Because it becomes the common API error, please check the correct procedure.
[5] XA_API_FATAL_INVALID_CMD_TYPE	0xFFFF8003	Abnormality has occurred, which disables process continuation. The subcommand was specified at the argument does not support. Because it becomes the common API error, please check the correct procedure.
[6] XA_TDM_CAP_EXEC_FATAL_STATE	0xFFFF90C0	Abnormality has occurred, which disables process continuation. The command does not follow procedure. Because it becomes the common API error, please check the correct procedure.
[8] XA_TDM_CAP_EXEC_FATAL_INTERNAL	0xFFFF90C1	Abnormality has occurred, which disables process continuation. Some of setting becomes incorrect after combination (out of memory, hardware module not available...). Because it becomes the common API error, please check the correct parameters and make sure the resource is validity.

[9] XA_TDM_CAP_CONFIG_FATAL_STATE	0xFFFF88C0	Abnormality has occurred, which disables process continuation. The command does not follow procedure. Because it becomes the common API error, please check the correct procedure.
[10] XA_TDM_CAP_CONFIG_FATAL_PCM_WIDTH	0xFFFF88C1	It is an error for TDM Capture specifications out of the range. The pcm width value was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16)
[11] XA_TDM_CAP_CONFIG_FATAL_CHANNEL_MODE	0xFFFF88C2	It is an error for TDM Capture specifications out of the range. The channel mode value was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16)
[12] XA_TDM_CAP_CONFIG_FATAL_SAMPLE_RATE	0xFFFF88C3	It is an error for TDM Capture specifications out of the range. The sample rate value was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16)
[13] XA_TDM_CAP_CONFIG_FATAL_FRAME_SIZE	0xFFFF88C4	It is an error for TDM Capture specifications out of the range. The frame size was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16)
[12] XA_TDM_CAP_CONFIG_FATAL_INVALID_INPUT	0xFFFF88C5	It is an error for TDM Capture specifications out of the range. The input value was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16)
[13] XA_TDM_CAP_CONFIG_FATAL_DMACHANNEL	0xFFFF88C6	It is an error for TDM Capture specifications out of the range. The adma channel was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16)
[14] XA_TDM_CAP_CONFIG_FATAL_VOLUME_RATE	0xFFFF88C7	It is an error for TDM Capture specifications out of the range. The volume rate was specified at the argument does not support. Please set an appropriate value.(Refer to 2.2.2.16)
[15]	Others	Reservered

3. Processing Flow

Figure 3-1 shows a flow diagram of processing performed by an application which uses this software. It applies for both case: TDM renderer and TDM capture.
The basic steps executed by the framework are white. The steps defined by the user framework are shaded. Design the process to suit the target system.

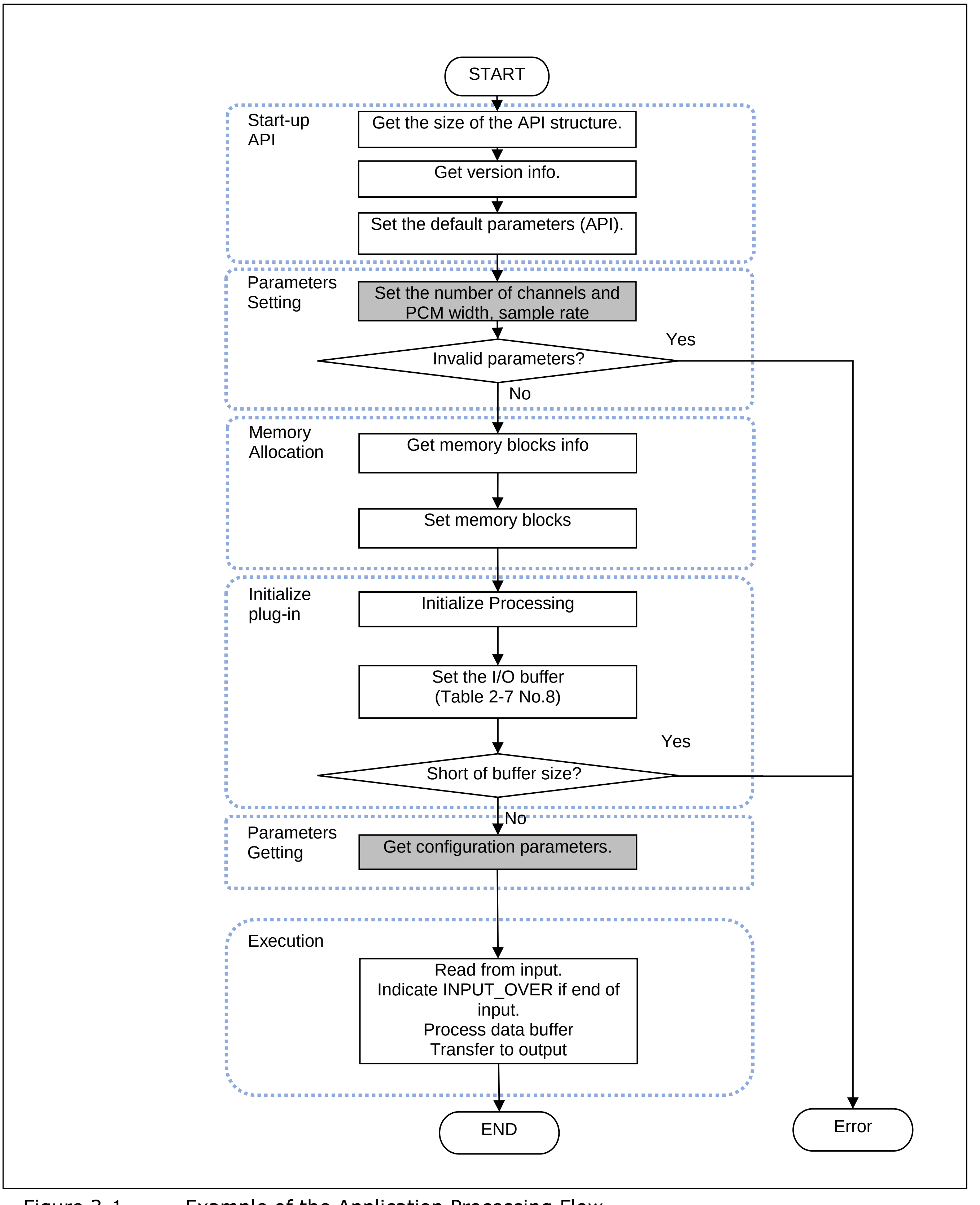

START
Start-up API
Get the size of the API structure.
Get version info.
Set the default parameters (API).
Parameters Setting
Set the number of channels and PCM width, sample rate
Invalid parameters?
Yes
No
Memory Allocation
Get memory blocks info
Set memory blocks
Initialize plug-in
Initialize Processing
Set the I/O buffer (Table 2-7 No.8)
Short of buffer size?
Yes
No
Parameters Getting
Get configuration parameters.
Execution
Read from input. Indicate INPUT_OVER if end of input. Process data buffer Transfer to output
END
Error

Figure 3-1 Example of the Application Processing Flow

4. Appendix

Below matrix tables show behavior of TDM plugin when user sets different sampling rate (Fs) (Hz) to plugin.

Table 4-1 Matrix table for sampling rate setting of TDM Renderer

Input Fs / Output Fs	32000	44100	48000
32000	-	-	-
44100	○	○	○
48000	○	○	○
0 (Non-use SRC)	*	○	○

Table 4-2 Matrix table for sampling rate setting of TDM Capture

Output Fs / Input Fs	32000	44100	48000
32000	-	-	-
44100	○	○	○
48000	○	○	○
0 (Non-use SRC)	*	○	○

○ : Plugin runs as normal

- : Plugin returns error due to invalid sample rate setting

* : Plugin enables SRC module automatically to perform sample rate conversion

Revision History	ADSP TDM Renderer/Capture Plugin User's Manual

Rev.	Date	Description	
		Page	Summary
1.00	Jan. 29, 2018	-	New Create
1.01	Jun. 28, 2018	-	Style Modify
1.02	Oct. 29, 2018	4	Add table 1-3 Support TDM Capture function Specification
2.00	Dec. 25, 2018	-	Official Release
		5	Update memory size

ADSP TDM Renderer/Capture Plugin User's Manual

Publication Date: Dec 25, 2018 Rev. 2.00

Published by: Renesas Electronics Corporation

RENESAS

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

http://www.renesas.com

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：https://www.renesas.com/contact/

Colophon 3.1

ADSP TDM Renderer/Capture Plugin

RCG3AHPLN0201ZDO

Renesas Electronics Corporation