**Document Type: Detail Design**

**Document Name:**

**ADSP FRAMEWORK: RENDERER CLASS**

Renesas Electronics Corporation

Renesas Design Vietnam Co. Ltd

R-Car Software Solution 2 Group

Middleware 1 Team

Table of Contents

[1. Overview 3](#_Toc534270654)

[2. Function list 4](#_Toc534270655)

[3. Detail information 5](#_Toc534270656)

[3.1 Data type and Macro definition 5](#_Toc534270657)

[3.2 Globle variable 6](#_Toc534270658)

[3.3 Function definition 6](#_Toc534270659)

[3.3.1 xa\_renderer\_prepare\_runtime 6](#_Toc534270660)

[3.3.2 xa\_renderer\_empty\_this\_buffer 8](#_Toc534270661)

[3.3.3 xa\_renderer\_fill\_this\_buffer 10](#_Toc534270662)

[3.3.4 xa\_renderer\_memtab 11](#_Toc534270663)

[3.3.5 xa\_renderer\_preprocess 13](#_Toc534270664)

[3.3.6 xa\_renderer\_postprocess 15](#_Toc534270665)

[3.3.7 xa\_renderer\_flush 17](#_Toc534270666)

[3.3.8 xa\_renderer\_cleanup 18](#_Toc534270667)

[3.3.9 xa\_renderer\_destroy 20](#_Toc534270668)

[3.3.10 xa\_renderer\_factory 21](#_Toc534270669)

[4. Revision history 22](#_Toc534270670)

List of Figures

[Figure 1‑1 The software architecture 3](#_Toc534270671)

[Figure 3‑1 xa\_renderer\_prepare\_runtime flowchart 7](#_Toc534270672)

[Figure 3‑2 xa\_renderer\_empty\_this\_buffer flowchart 9](#_Toc534270673)

[Figure 3‑3 xa\_renderer\_fill\_this\_buffer flowchart 10](#_Toc534270674)

[Figure 3‑4 xa\_renderer\_memtab flowchart 12](#_Toc534270675)

[Figure 3‑5 xa\_renderer\_preprocess flowchart 14](#_Toc534270676)

[Figure 3‑6 xa\_renderer\_postprocess flowchart 16](#_Toc534270677)

[Figure 3‑7 xa\_renderer\_flush flowchart 17](#_Toc534270678)

[Figure 3‑8 xa\_renderer\_cleanup flowchart 19](#_Toc534270679)

[Figure 3‑9 xa\_renderer\_destroy flowchart 20](#_Toc534270680)

[Figure 3‑10 xa\_renderer\_factory flowchart 21](#_Toc534270681)

List of Table

[Table 2‑1 Function list 4](#_Toc534270682)

[Table 3‑1 XARenderer type structure information 5](#_Toc534270683)

[Table 3‑2 Macro definitions 5](#_Toc534270684)

[Table 3‑3 Global variable xa\_renderer\_cmd 6](#_Toc534270685)

# Overview

This section provides an overview of software architecture..

User Application

ADSP Interface

Kernel Space

User Space

ADSP Driver

ARM

Audio HW

**ADSP Framework**

TDM class

Equalizer class

Capture class

Renderer class

ADSP

Plugin

Equalizer Plugin\*

TDM Plugin

Capture Plugin

Renderer Plugin

This document’s target is in side of red square

\* not connect to SCU/SSI ADMA block

SCU/ SSI/

ADMA

DAC/

ADC

Figure 1‑1 ****The software architecture****

# Function list

The following is list of functions:

Table 2‑1 Function list

|  |  |  |
| --- | --- | --- |
| **Type** | **Function Name** | **Outline** |
| API | xa\_renderer\_factory | This API is to construct generic audio component and set functions pointer for internal functions. |
| Internal functions | xa\_renderer\_prepare\_runtime | This function is to prepare renderer for steady operation. |
| xa\_renderer\_empty\_this\_buffer | This function is to put message into input port queue. |
| xa\_renderer\_fill\_this\_buffer | This function is dummy function to send respond message to user. |
| xa\_renderer\_flush | This function is to cancel data processing and purge input port queue. |
| xa\_renderer\_memtab | This function is to initialize input port structure. |
| xa\_renderer\_preprocess | This function is to prepare data for input buffer. |
| xa\_renderer\_postprocess | This function is to consume bytes from input buffer. |
| xa\_renderer\_terminate | This function is to check if it received internal message. |
| xa\_renderer\_destroy | This function is to destroy input port and renderer component. |
| xa\_renderer\_cleanup | This function is to purge input port and cancel data processing. |

# Detail information

This section describes detail information of data types, macro definitions, implemented APIs and internal function units, globle variable.

## Data type and Macro definition

The XARenderer type structure is the work area used by the renderer class. When using this class, 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 class. Make sure not to change the value of this area with the application program.

Table 3‑1 XARenderer type structure information

|  |  |
| --- | --- |
| Member name | Outline |
| XACodecBase base | Generic audio codec data |
| xf\_input\_port\_t input | Input port queue |
| xf\_output\_port\_t output | Output port queue |
| xa\_renderer\_cb\_t cdata | Buffer completion hook |
| xf\_message\_t msg | Output ready message |
| u32 factor | Time conversion factor (input byte "duration" in timebase units) |
| u32 schedule | Internal message scheduling flag (shared with interrupt) |

Table 3‑2 Macro definitions

|  |  |  |
| --- | --- | --- |
| Macro | Value | Outline |
| XA\_RENDERER\_FLAG\_ RUNNING | \_\_XA\_BASE\_FLAG(1 << 0) | Rendering is performed |
| XA\_RENDERER\_FLAG\_SILENCE | \_\_XA\_BASE\_FLAG(1 << 1) | Renderer is idle and produces silence |
| XA\_RENDERER\_FLAG\_OUTPUT\_READY | \_\_XA\_BASE\_FLAG(1 << 2) | Ouput data is ready |

Note: Macro \_\_XA\_BASE\_FLAG(f) ((f) << 6) is in xa-class-base.h file.

## Globle variable

Table 3‑3 Global variable xa\_renderer\_cmd

|  |  |
| --- | --- |
| static XA\_ERRORCODE (\* const xa\_renderer\_cmd[])(XACodecBase \*base, xf\_message\_t \*m) | |
| Description: variable stores function pointers according to opcode index to run in runtime operation. | |
| Array index | Value (function pointer) |
| XF\_OPCODE\_TYPE(XF\_SET\_PARAM) | xa\_base\_set\_param |
| XF\_OPCODE\_TYPE(XF\_GET\_PARAM) | xa\_base\_get\_param |
| XF\_OPCODE\_TYPE(XF\_EMPTY\_THIS\_BUFFER) | xa\_renderer\_empty\_this\_buffer |
| XF\_OPCODE\_TYPE(XF\_FILL\_THIS\_BUFFER) | xa\_renderer\_fill\_this\_buffer |
| XF\_OPCODE\_TYPE(XF\_FLUSH) | xa\_renderer\_flush |

Note: Marco XF\_OPCODE\_TYPE(opcode) : ((opcode) & (0x3F)) in xf-opcode.h file.

## Function definition

### xa\_renderer\_prepare\_runtime

DD\_FWK\_RDR\_02\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static inline XA\_ERRORCODE xa\_renderer\_prepare\_runtime(XARenderer \*renderer) | | | |
| **Function** | This function is to prepare renderer for steady operation. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XARenderer | renderer | I/O | Pointer to codec instance structure (struct XARenderer) |
| **Return value** | XA\_API\_FATAL\_MEM\_ALLOC | | API structure is NULL (error from plugin). | |
| XA\_API\_FATAL\_MEM\_ALIGN | | API structure is not aligned to 4 bytes (error from plugin). | |
| XA\_RDR\_CONFIG\_FATAL\_STATE | | Pre-initialization is not completed yet (error from plugin). | |
| XA\_RDR\_CONFIG\_FATAL\_RANGE | | Sample rate is invalid. | |
| XA\_NO\_ERROR | | Normally end. | |
| **Description** | * xa\_renderer\_prepare\_runtime command processing:   - Get config parameters: sample rate, pcm width and channels.  - Set renderer timestamp factor. | | | |

[Covers: FD\_FWK\_CMN\_002]

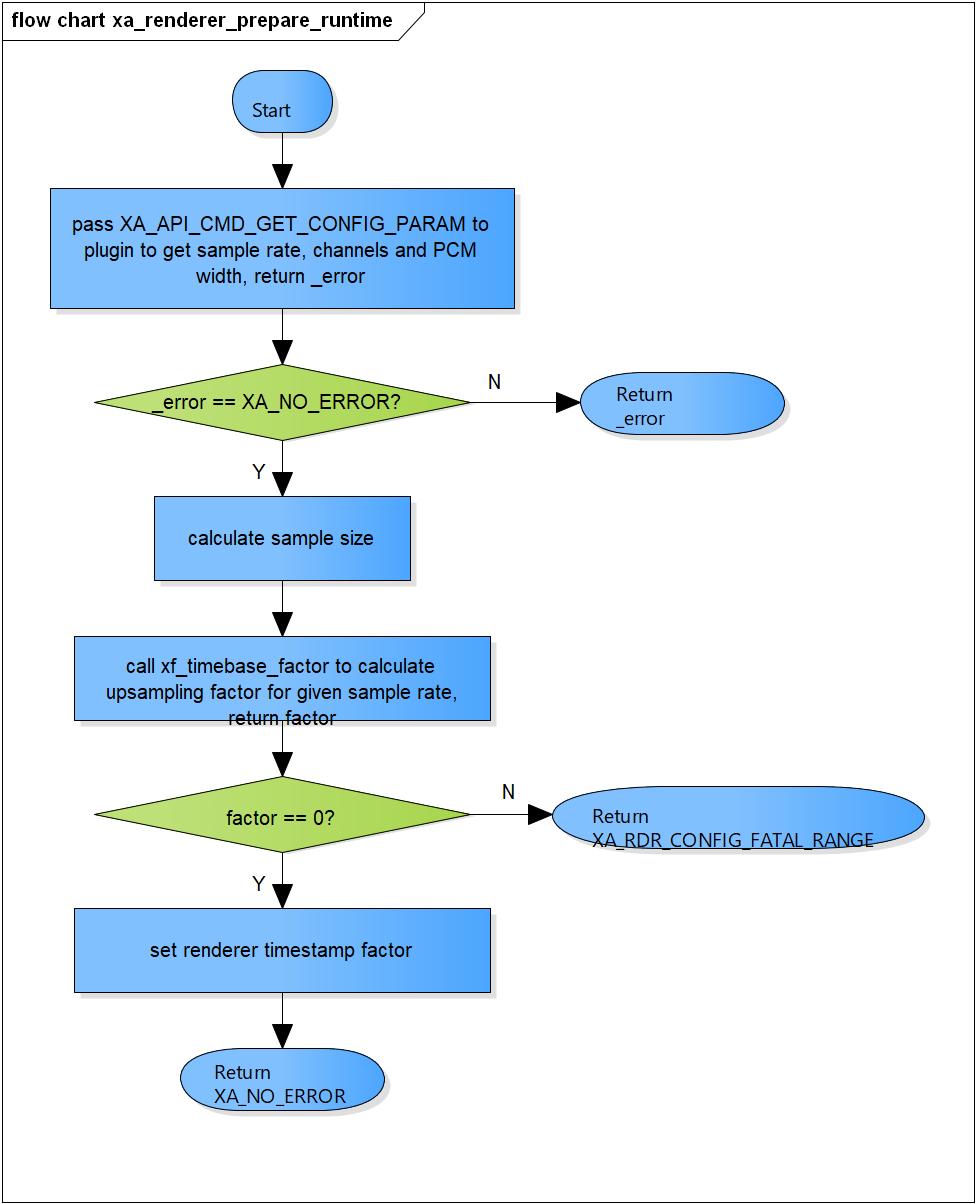


Figure 3‑1 xa\_renderer\_prepare\_runtime flowchart

### xa\_renderer\_empty\_this\_buffer

DD\_FWK\_RDR\_02\_002

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_renderer\_empty\_this\_buffer(XACodecBase \*base,  xf\_message\_t \*m) | | | |
| **Function** | This function is to put message into the input port. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I/O | Pointer to codec instance structure (struct XACodecBase) |
| xf\_message\_t | m | I | Pointer to audio message (struct xf\_message) |
| **Return value** | XA\_API\_FATAL\_INVALID\_CMD\_TYPE | | Port is not an input port. | |
| XA\_API\_FATAL\_INVALID\_CMD | | Post initialization state is not completed yet. | |
| XA\_NO\_ERROR | | Normally end. | |
| **Description** | * xa\_renderer\_empty\_this\_buffer command processing:   - Make sure the port is an input port.  - Check if post initialization state is not completed yet  - Put message into the input port. | | | |

[Covers: FD\_FWK\_CMN\_010, FD\_FWK\_CMN\_005]

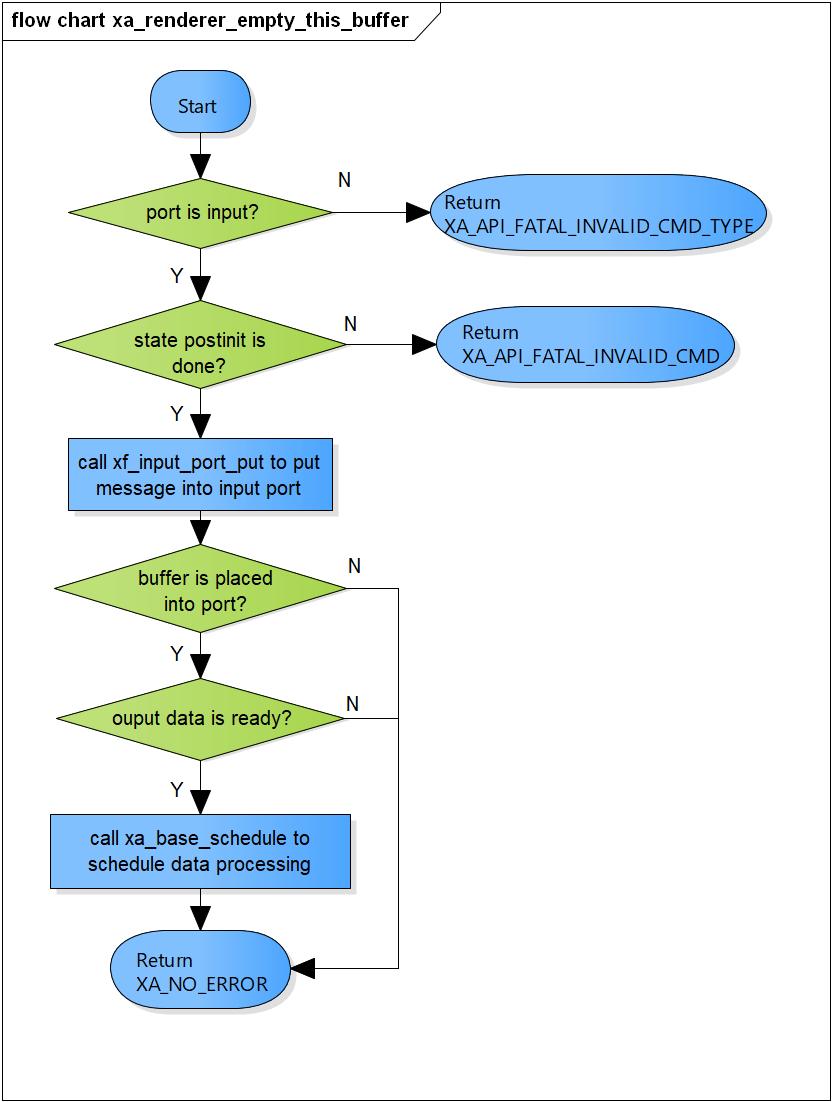


Figure 3‑2 xa\_renderer\_empty\_this\_buffer flowchart

### xa\_renderer\_fill\_this\_buffer

DD\_FWK\_RDR\_02\_003

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_renderer\_fill\_this\_buffer(XACodecBase \*base,  xf\_message\_t \*m) | | | |
| **Function** | This is a dummy function. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | x | Pointer to codec instance structure (struct XACodecBase). |
| xf\_message\_t | m | I/O | Pointer to audio message (struct xf\_message). |
| **Return value** | XA\_NO\_ERROR | | Normally end. | |
| **Description** | * xa\_renderer\_fill\_this\_buffer command processing:   - This is a dummy function send message to user. | | | |

[Covers: FD\_FWK\_CMN\_011, FD\_FWK\_CMN\_005]

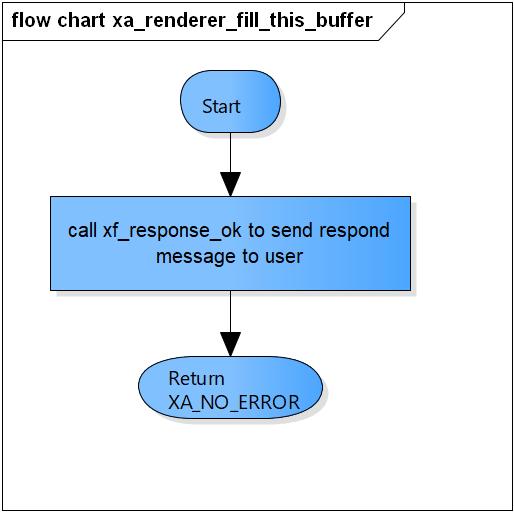


Figure 3‑3 xa\_renderer\_fill\_this\_buffer flowchart

### xa\_renderer\_memtab

DD\_FWK\_RDR\_02\_004

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_renderer\_memtab(XACodecBase \*base, WORD32 idx, WORD32 type, WORD32 size, WORD32 align, u32 core) | | | |
| **Function** | This function is to initialize input port structure. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I/O | Pointer to codec instance structure (struct XACodecBase). |
| WORD32 | idx | I | Port index  Valid value: 1 |
| WORD32 | type | I | Port type (input or output port)  Valid value: XA\_MEMTYPE\_INPUT |
| WORD32 | size | I | Buffer size of port |
| WORD32 | align | I | Port alignment |
| u32 | core | I | Core index of ADSP framework |
| **Return value** | XA\_NO\_ERROR | | Normally end. | |
| XA\_API\_FATAL\_INVALID\_CMD\_TYPE | | Port type is not an input port.  Port index are invalid.  idx is an invalid memory index (error from plugin). | |
| XA\_API\_FATAL\_MEM\_ALLOC | | Initialize input port is fail.  API structure is NULL (error from plugin). | |
| XA\_API\_FATAL\_MEM\_ALIGN | | API structure is not aligned to 4 bytes (error from plugin). | |
| XA\_CAP\_CONFIG\_FATAL\_STATE | | Post initialization state is not completed yet (error from plugin). | |
| **Description** | * xa\_renderer\_memtab command processing:   - Initialize input port structure.  - Mark renderer output buffer is ready. | | | |

[Covers: FD\_FWK\_CMN\_008, FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_001]

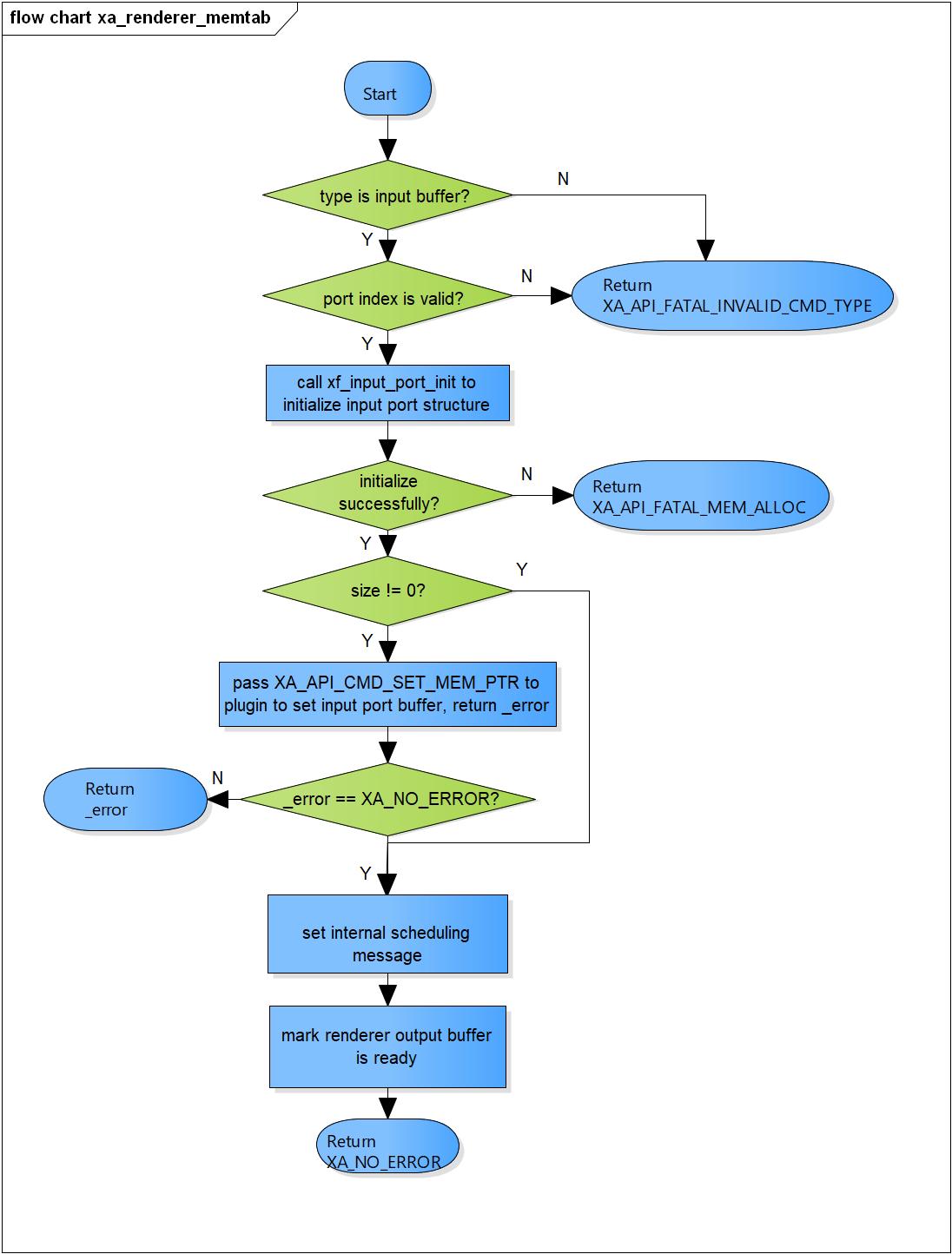


Figure 3‑4 xa\_renderer\_memtab flowchart

### xa\_renderer\_preprocess

DD\_FWK\_RDR\_02\_005

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_renderer\_preprocess(XACodecBase \*base) | | | |
| **Function** | This function is to prepare data for input buffer. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I | Pointer to codec instance structure (struct XACodecBase) |
| **Return value** | XA\_NO\_ERROR | | Normally end | |
| XA\_RDR\_EXEC\_NONFATAL\_INPUT | | No input data available.  Input data is insufficient. | |
| XA\_API\_FATAL\_MEM\_ALLOC | | API structure is NULL (error from plugin). | |
| XA\_API\_FATAL\_MEM\_ALIGN | | API structure is not aligned to 4 bytes (error from plugin). | |
| XA\_RDR\_EXEC\_FATAL\_STATE | | Initialization state is not completed yet (error from plugin). | |
| **Description** | * Preprocess command processing:   - Check current execution stage is runtime initialization.  - Check if input stream is over.  - Submit input buffer to the renderer.  - Set total number of bytes we have in buffer. | | | |

[Covers: FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_002, FD\_FWK\_CMN\_003]

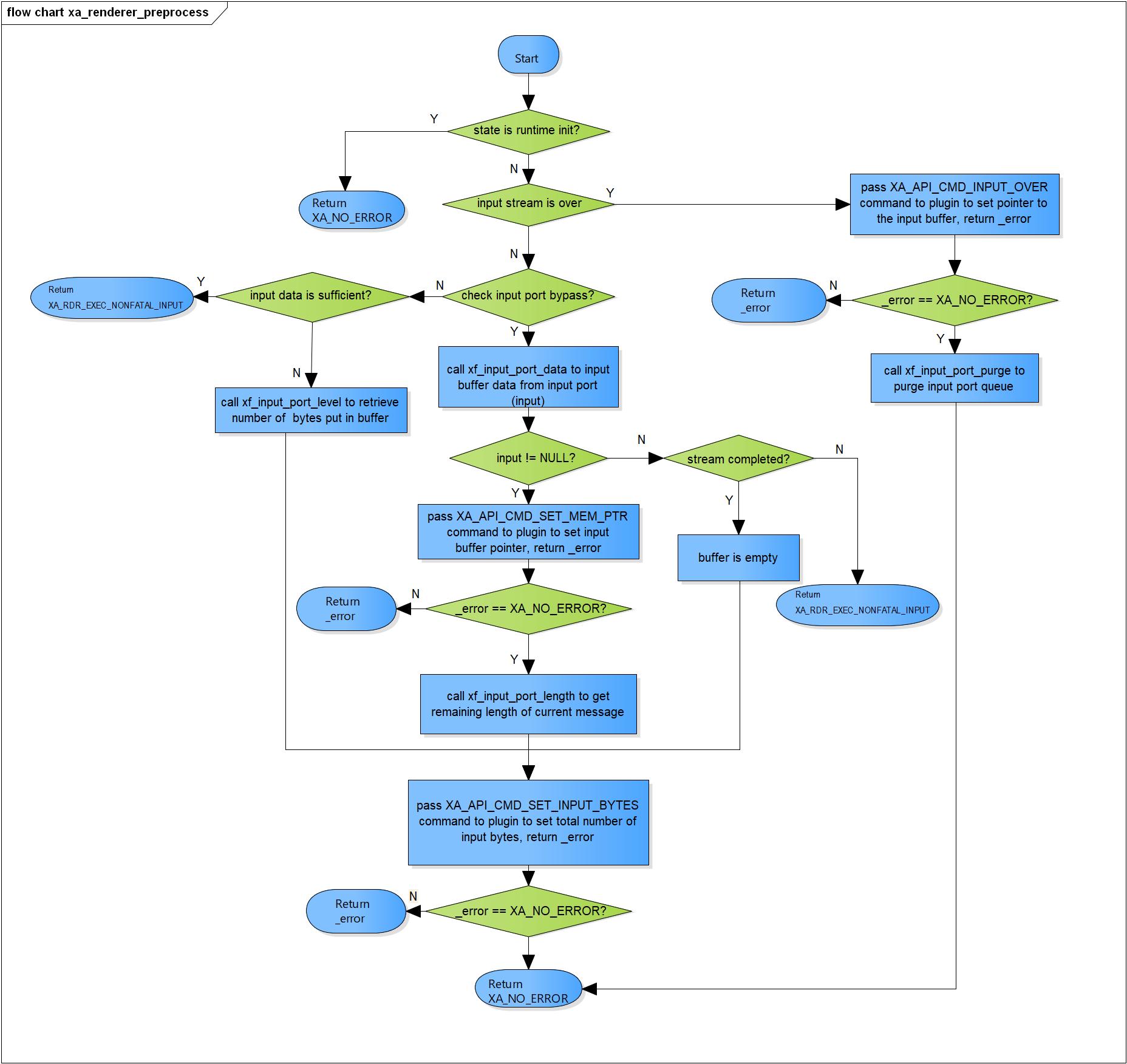


Figure 3‑5 xa\_renderer\_preprocess flowchart

### xa\_renderer\_postprocess

DD\_FWK\_RDR\_02\_006

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_renderer\_postprocess(XACodecBase \*base, s32 done) | | | |
| **Function** | This function is to consume bytes from input buffer. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I/O | Pointer to codec instance structure (struct XACodecBase). |
| s32 | done | I | State of processing of plugin. |
| **Return value** | XA\_NO\_ERROR | | Normally end. | |
| XA\_API\_FATAL\_MEM\_ALLOC | | API structure is NULL (error from plugin). | |
| XA\_API\_FATAL\_MEM\_ALIGN | | API structure is not aligned to 4 bytes (error from plugin). | |
| XA\_RDR\_EXEC\_FATAL\_STATE | | Input buffer is not exist (error from plugin). | |
| **Description** | * Postprocess command processing:   - Check state if process is done.  - Get total amount of consumed samples from plugin.  - Consume bytes from input buffer.  - Reschedule execution if we have pending input. | | | |

[Covers: FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_002, FD\_FWK\_CMN\_003]

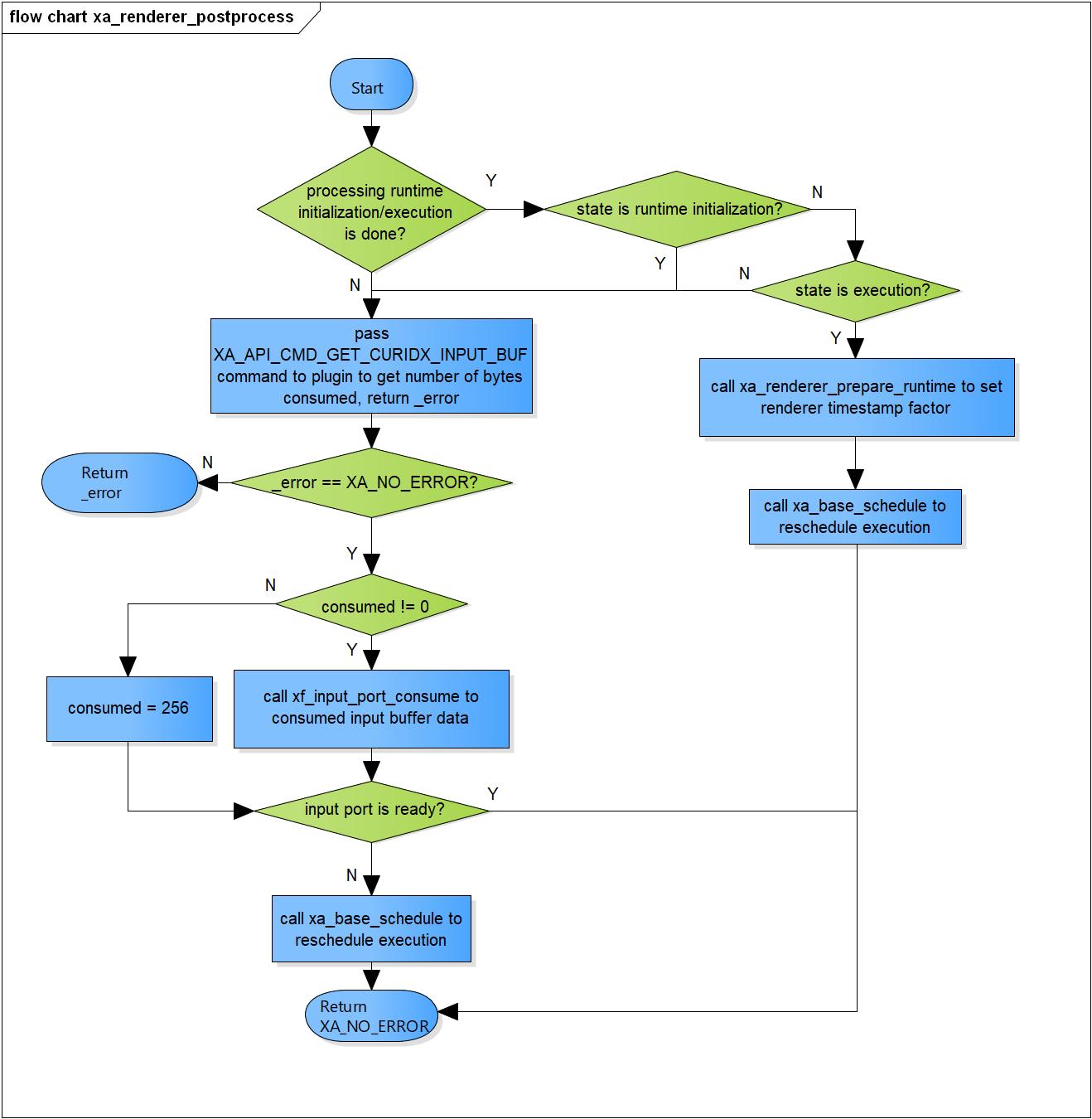


Figure 3‑6 xa\_renderer\_postprocess flowchart

### xa\_renderer\_flush

DD\_FWK\_RDR\_02\_007

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static XA\_ERRORCODE xa\_renderer\_flush(XACodecBase \*base, xf\_message\_t \*m) | | | |
| **Function** | This function is to cancel data processing and purge input port queue. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XACodecBase | base | I/O | Pointer to codec instance structure (struct XACodecBase). |
| xf\_message\_t | m | I | Pointer to audio message (struct xf\_message). |
| **Return value** | XA\_NO\_ERROR | | Normally end. | |
| XA\_API\_FATAL\_INVALID\_CMD | | Processing is not in execution state. | |
| XA\_API\_FATAL\_INVALID\_CMD\_TYPE | | Message length is not zero. | |
| **Description** | * xa\_renderer\_flush command processing:   - Check if process is in execution state.  - Ensure input parameter length is zero.  - Cancel data processing if needed.  - Input port flushing; purge content of input buffer. | | | |

[Covers: FD\_FWK\_CMN\_012, FD\_FWK\_CMN\_005]

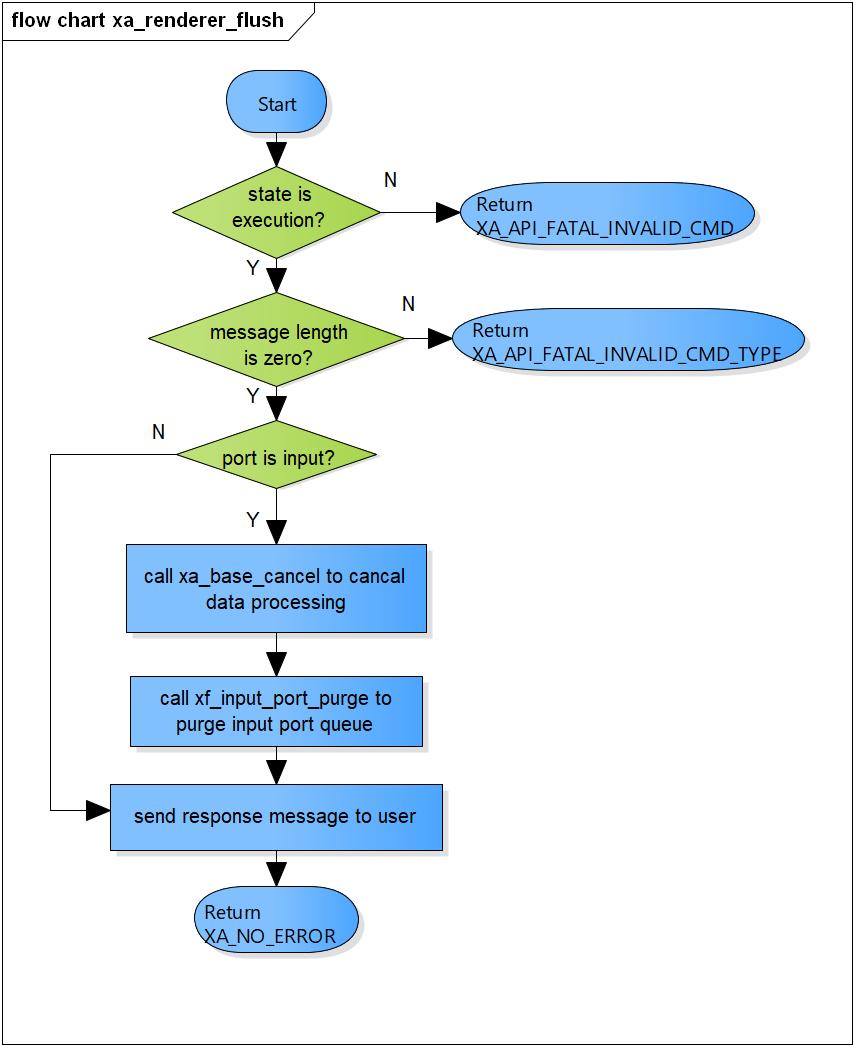


Figure 3‑7 xa\_renderer\_flush flowchart

### xa\_renderer\_cleanup

DD\_FWK\_RDR\_02\_008

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static s32 xa\_renderer\_cleanup(xf\_component\_t \*component, xf\_message\_t \*m) | | | |
| **Function** | This function is to purge input port and cancel data processing. | | | |
| **Arguments** | Type | Name | I/O | Description |
| xf\_component\_t | component | I/O | Pointer to codec instance structure (struct xf\_component ). |
| xf\_message\_t | m | I | Pointer to audio message  (struct xf\_message). |
| **Return value** | 1 | | Renderer component cannot be destroyed. | |
| 0 | | Renderer component has been destroyed totally. | |
| **Description** | * xa\_renderer\_cleanup command processing:   - Complete message with error result code.  - Cancel component task execution if needed.  - Pass XA\_CAP\_CONFIG\_PARAM\_STATE command to plugin with state is idle.  - Purge input port.  - Check if we have internal message scheduled. | | | |

[Covers: FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_004]

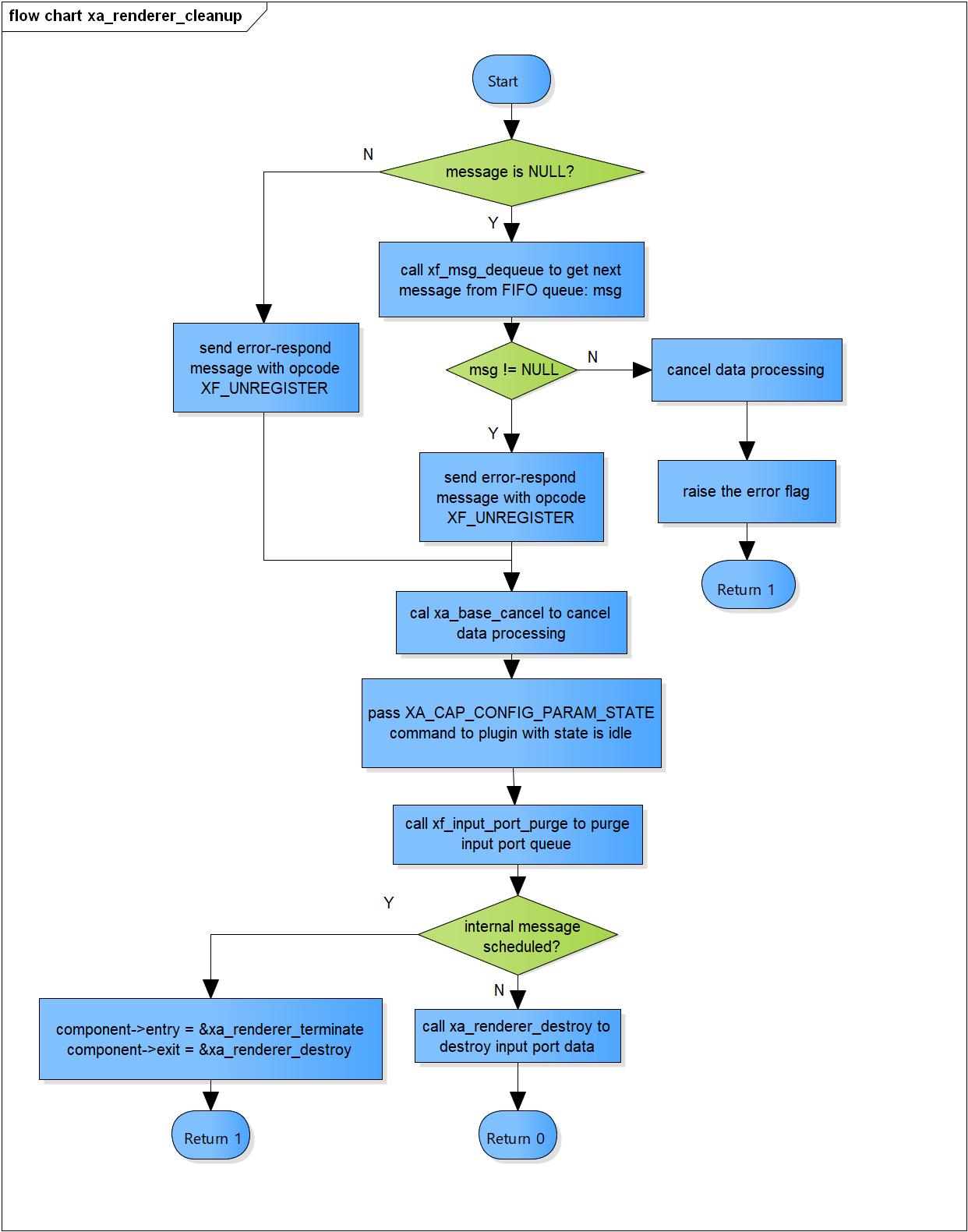


Figure 3‑8 xa\_renderer\_cleanup flowchart

### xa\_renderer\_destroy

DD\_FWK\_RDR\_02\_009

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static s32 xa\_renderer\_destroy(xf\_component\_t \*component, xf\_message\_t \*m) | | | |
| **Function** | This function is to destroy input port and renderer component. | | | |
| **Arguments** | Type | Name | I/O | Description |
| xf\_component\_t | component | I | Pointer to codec instance structure (struct xf\_component). |
| xf\_message\_t | m | x | Pointer to audio message (struct xf\_message). |
| 0 | | Indicate the component is destroyed. | |
| **Description** | * xa\_renderer\_terminate command processing:   - Destroy input port.  - Destroy base object. | | | |

[Covers: FD\_FWK\_CMN\_004]

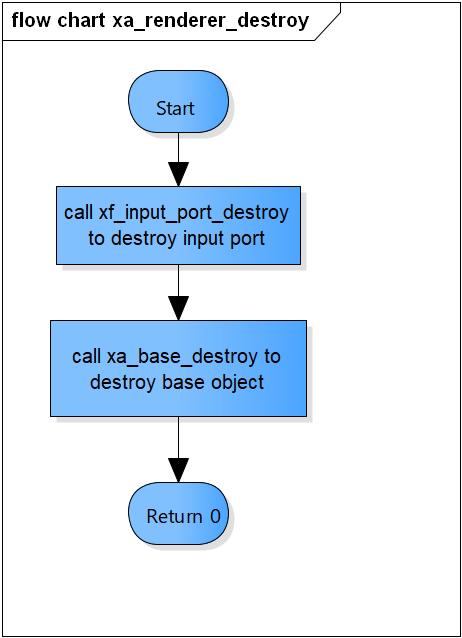


Figure 3‑9 xa\_renderer\_destroy flowchart

### xa\_renderer\_factory

DD\_FWK\_RDR\_02\_010

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | xf\_component\_t \* xa\_renderer\_factory(u32 core, xa\_codec\_func\_t process) | | | |
| **Function** | This function is to initialize Renderer component. | | | |
| **Arguments** | Type | Name | I/O | Description |
| u32 | core | I | Core index of ADSP framework. |
| xa\_codec\_func\_t | process | I | Codec API entry point (function). |
| **Return Value** | Return handle to component. | | | |
| **Description** | * xa\_renderer\_factory command processing:   - Initialize Renderer component. | | | |

[Covers: FD\_FWK\_CMN\_005, FD\_FWK\_CMN\_008, FD\_FWK\_CMN\_009]

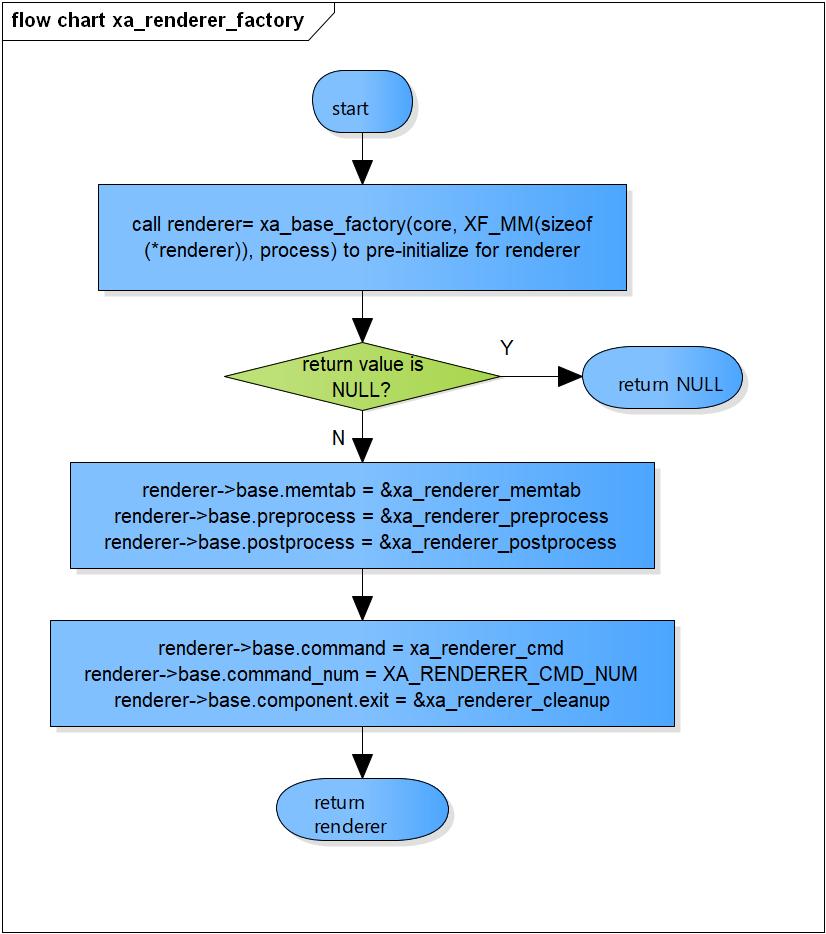


Figure 3‑10 xa\_renderer\_factory flowchart

# Revision history

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Page** | **Content** | **Approved** | **Changed** |
| 1.0.0 | Nov 14 2018 | - | First Edition issued | Vu Phan | Vu Phan |
| 1.1.0 | Dec 10 2018 | - | Add traceability ID | Vu Phan | Ngu Pham |
| 1.2.0 | Jan 03, 2019 | - | Add range for input parameters | Vu Phan | Tien Tran |