**Document Type: Detail Design**

**Document Name:**

**ADSP OMX RDR**

Renesas Design Vietnam Co. Ltd

R-Car Software Solution 2 Group

Middleware 1 Team

Renesas Electronics Corporation

Table of Contents

[1. Overview 3](#_Toc534269798)

[2. Function list 4](#_Toc534269799)

[3. Detail information 5](#_Toc534269800)

[3.1 Data type and Macro definition 5](#_Toc534269801)

[3.2 Function definition 6](#_Toc534269802)

[3.2.1 EQUALIZER\_Setup 6](#_Toc534269803)

[3.2.2 EQUALIZER\_RuntimeInit 7](#_Toc534269804)

[3.2.3 EQUALIZER\_GetParam 8](#_Toc534269805)

[3.2.4 EQUALIZER\_TimeStamp 9](#_Toc534269806)

[3.2.5 EQUALIZER\_GetParameter 10](#_Toc534269807)

[3.2.6 EQUALIZER\_SetParameter 11](#_Toc534269808)

[3.2.7 EQUALIZER\_ComponentInit 12](#_Toc534269809)

[3.2.8 EQUALIZER\_ComponentDeInit 13](#_Toc534269810)

[3.2.9 EQUALIZER\_ComponentCreate 14](#_Toc534269811)

[4. Revision history 16](#_Toc534269812)

List of Figures

[Figure 1‑1 The software architecture 3](#_Toc534269813)

[Figure 3‑1 EQUALIZER\_Setup flowchart 6](#_Toc534269814)

[Figure 3‑2 EQUALIZER\_RuntimeInit flowchart 7](#_Toc534269815)

[Figure 3‑3 EQUALIZER\_GetParam flowchart 8](#_Toc534269816)

[Figure 3‑4 EQUALIZER\_TimeStamp flowchart 9](#_Toc534269817)

[Figure 3‑5 EQUALIZER\_GetParameter flowchart 10](#_Toc534269818)

[Figure 3‑6 EQUALIZER\_SetParameter flowchart 11](#_Toc534269819)

[Figure 3‑7 EQUALIZER\_ComponentInit flowchart 12](#_Toc534269820)

[Figure 3‑8 EQUALIZER\_ComponentDeInit flowchart 13](#_Toc534269821)

[Figure 3‑9 EQUALIZER\_ComponentCreate flowchart 15](#_Toc534269822)

List of Table

[Table 2‑1 Function list of OMX Equalizer 4](#_Toc534269823)

[Table 3‑1 XAOMXEqualizer type structure information 5](#_Toc534269824)

[Table 3‑2 Macro definitions of OMX Equalizer 5](#_Toc534269825)

# Overview

In this chapter, overview of OMX Equalizer interface is explained.

User Application

OMX TDM

OMX Renderer/Capture

OMX Equalizer

User Space

ADSP Interface for Linux (base)

Kernel Space

ARM

ADSP Driver for Linux

Audio HW

ADSP

NC/EC

ADSP Framework

Renderer/Capture Plugin

TDM Plugin

Equalizer Plugin

Codec

This document’s target is in side of red square.

DAC/ADC

SRC/CTU/MIX/ DVC/ SSI/

ADMA

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

# Function list

The following is list of functions:

Table 2‑1 Function list of OMX Equalizer

|  |  |  |
| --- | --- | --- |
| **No.** | **Function Name** | **Outline** |
| 1 | EQUALIZER\_Setup | Equalizer prepare codec setup parameters. |
| 2 | EQUALIZER\_RuntimeInit | Return no parameter is set in runtime. |
| 3 | EQUALIZER\_GetParam | Check the message length is 0. |
| 4 | EQUALIZER\_TimeStamp | Calculate timestamp for current output buffer and next buffer. |
| 5 | EQUALIZER\_GetParameter | Get Equalizer parameter. |
| 6 | EQUALIZER\_SetParameter | Set Equalizer parameter. |
| 7 | EQUALIZER\_ComponentDeInit | Destroy Equalizer component and private data. |
| 8 | EQUALIZER\_ComponentInit | Initialize base codec interface, parameters, PCM and Equalizer format. |
| 9 | EQUALIZER\_ComponentCreate | Create base component and initialize the Equalizer component. |

# Detail information

This section describes detail information of data types, macro definitions.

## Data type and Macro definition

Table 3‑1 XAOMXEqualizer type structure information

|  |  |
| --- | --- |
| Member name | Outline |
| XAOMXCodecBase base | Generic codec structure. |
| OMX\_AUDIO\_PARAM\_PCMMODETYPE sPCM\_in | PCM-specific parameters (input port). |
| OMX\_AUDIO\_PARAM\_PCMMODETYPE sPCM\_out | PCM-specific parameters (output port). |
| XAOMX\_AUDIO\_PARAM\_EQUALIZER sEQU | Equalizer parameters. |

Note:

Struct XAOMXCodecBase is a base codec structure of OMX in omx-codec-base.h

Struct OMX\_AUDIO\_PARAM\_PCMMODETYPE is a PCM format description in OMX\_Audio.h

Struct XAOMX\_AUDIO\_PARAM\_EQUALIZER is a parametric Equalizer settings structure.

Table 3‑2 Macro definitions of OMX Equalizer

|  |  |  |
| --- | --- | --- |
| Macro | Value | Outline |
| NUM\_INPUT\_BUFFERS | 4 | Total amount of input buffers. |
| INPUT\_BUFFER\_LENGTH | 4096 | Default input buffer length. |
| NUM\_OUTPUT\_BUFFERS | 4 | Total amount of output buffers. |
| OUTPUT\_BUFFER\_LENGTH | 4096 | Default output buffer length. |
| BUFFER\_ALIGNMENT | 32 | Required data alignment of Equalizer. |

## Function definition

### EQUALIZER\_Setup

DD\_API\_EQZ\_01\_001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static s32 EQUALIZER\_Setup(XAOMXCodecBase\*pBase, xf\_set\_param\_msg\_t \*msg) | | | |
| **Function** | Equalizer prepare codec setup parameters. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XAOMXCodecBase\* | pBase | I | Pointer to base codec structure of Equalizer component. |
| xf\_set\_param\_msg\_t\* | msg | O | Message send to Equalizer plugin. |
| **Return value** | length | | Number of parameters want to set for Equalizer plugin. | |
| **Description** | * EQUALIZER\_Setup processing:   - Prepare PCM width, sample rate, Equalizer type, center frequencies, filter type, filter bandwidth, filter gain, and filter gain base values into command message. | | | |

[Covers: FD\_API\_EQZ\_003]

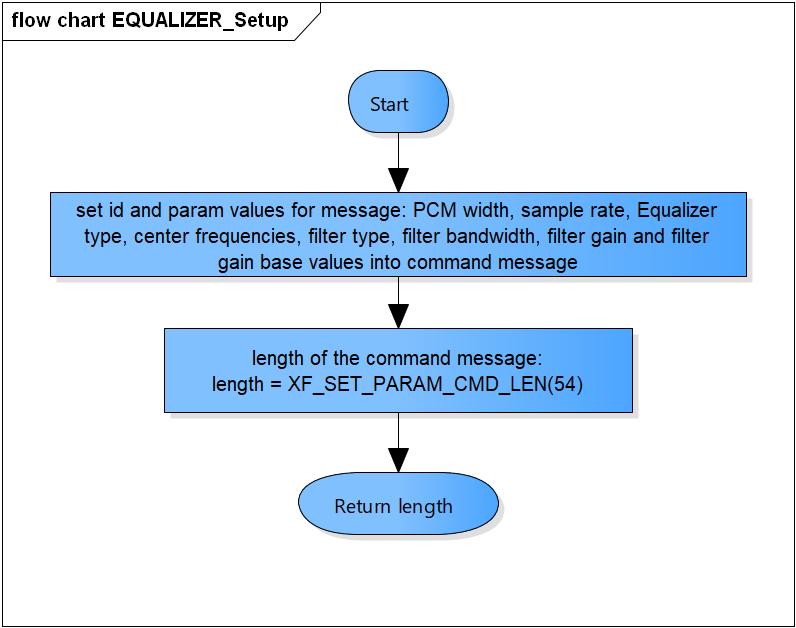


Figure 3‑1 EQUALIZER\_Setup flowchart

### EQUALIZER\_RuntimeInit

DD\_API\_EQZ\_01\_002

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static s32 EQUALIZER\_RuntimeInit(XAOMXCodecBase\*pBase, xf\_start\_msg\_t \*msg) | | | |
| **Function** | Return no parameter is set in runtime. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XAOMXCodecBase\* | pBase | I | Pointer to base codec structure of Equalizer component. |
| xf\_start\_msg\_t\* | msg | O | Message send to Equalizer plugin. |
| **Return value** | 0 | | No parameter is set in runtime | |
| **Description** | * EQUALIZER\_RuntimeInit processing:   - Return no parameter is set in runtime. | | | |

[Covers: FD\_API\_EQZ\_003]

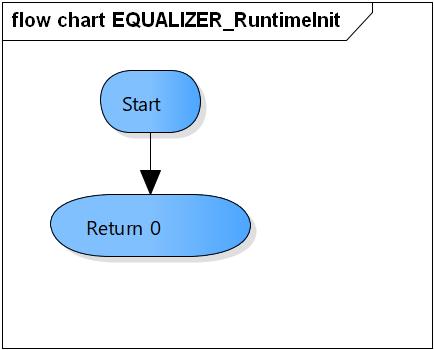


Figure 3‑2 EQUALIZER\_RuntimeInit flowchart

### EQUALIZER\_GetParam

DD\_API\_EQZ\_01\_003

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static s32 EQUALIZER\_GetParam(XAOMXCodecBase \*pBase,  xf\_get\_param\_msg\_t \*msg,  u32 length) | | | |
| **Function** | Check the message length is 0. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XAOMXCodecBase\* | pBase | I | Pointer to base codec structure of Equalizer component. |
| xf\_get\_param\_msg\_t \* | msg | I | Message receive from Equalizer plugin. |
| u32 | length | I | Length of message. |
| **Return value** | 0 | | Normal return. | |
| -EBADF | | The message length is invalid. | |
| **Description** | * EQUALIZER\_GetParam processing:   - Check the message length is 0. | | | |

[Covers: FD\_API\_EQZ\_003]

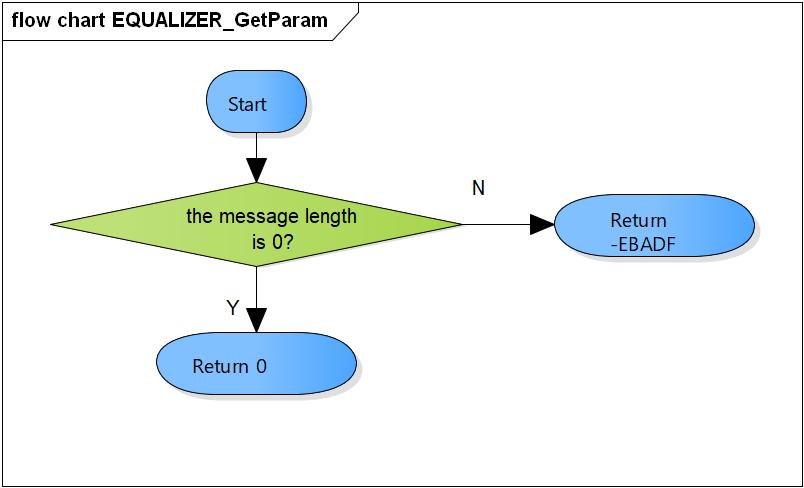


Figure 3‑3 EQUALIZER\_GetParam flowchart

### EQUALIZER\_TimeStamp

DD\_API\_EQZ\_01\_004

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static void EQUALIZER\_TimeStamp(XAOMXCodecBase \*pBase,  OMX\_BUFFERHEADERTYPE \*pBufHdr) | | | |
| **Function** | Calculate timestamp for current output buffer and next buffer. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XAOMXCodecBase\* | pBase | I/O | Pointer to base codec structure of Equalizer component. |
| OMX\_BUFFERHEADERTYPE\* | msg | I | Output buffer. |
| **Return value** | None. | | | |
| **Description** | * EQUALIZER\_TimeStamp processing:   - Add current timestamp to the output buffer.  - Calculate timestamp for the next buffer. | | | |

[Covers: FD\_API\_EQZ\_003]

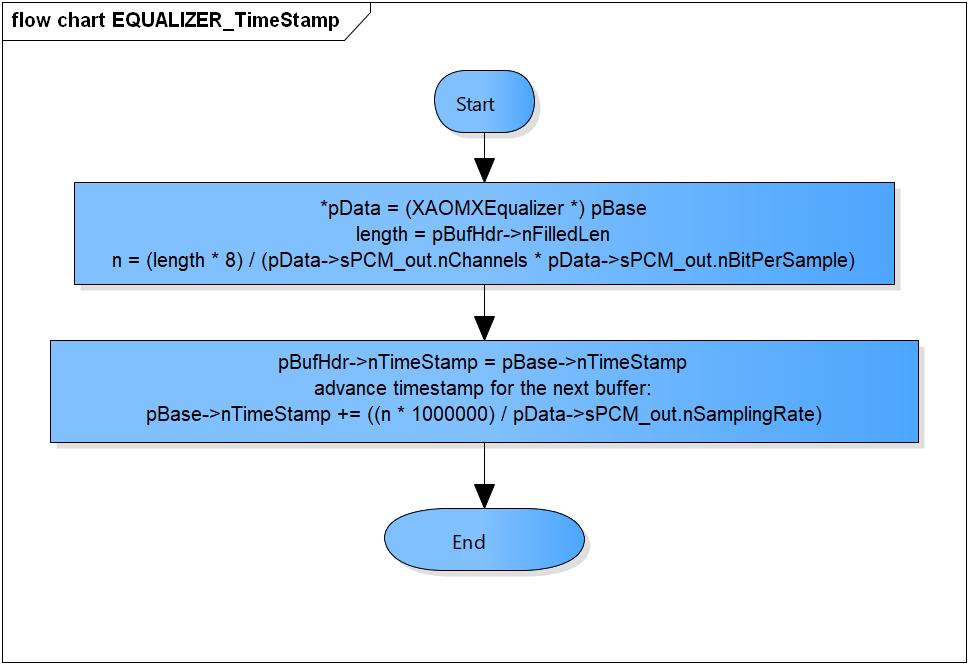


Figure 3‑4 EQUALIZER\_TimeStamp flowchart

### EQUALIZER\_GetParameter

DD\_API\_EQZ\_01\_005

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static OMX\_ERRORTYPE EQUALIZER\_GetParameter(XAOMXCodecBase \*pBase,  OMX\_INDEXTYPE nIndex,  OMX\_PTR pParam) | | | |
| **Function** | Get parameter and save to pParam. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XAOMXCodecBase\* | pBase | I | Pointer to base codec structure of Equalizer component. |
| OMX\_INDEXTYPE | nIndex | I | Index of parameter structure.  Supported range:  OMX\_IndexParamAudioPcm  XAOMX\_IndexParamAudioEqualizer  OMX\_IndexParamCompBufferSupplier |
| OMX\_PTR | pParam | O | Pointer to parameter structure want to save. |
| **Return value** | OMX\_ErrorNone | | Normal return. | |
| OMX\_ErrorUnsupportedIndex | | Wrong Index of structure. | |
| **Description** | * EQUALIZER\_GetParameter processing:   - Get parameters of OMX\_AUDIO\_PARAM\_PCMMODETYPE, XAOMX\_AUDIO\_PARAM\_EQUALIZER, OMX\_PARAM\_BUFFERSUPPLIERTYPE structure base on nIndex. | | | |

[Covers: FD\_API\_EQZ\_004]

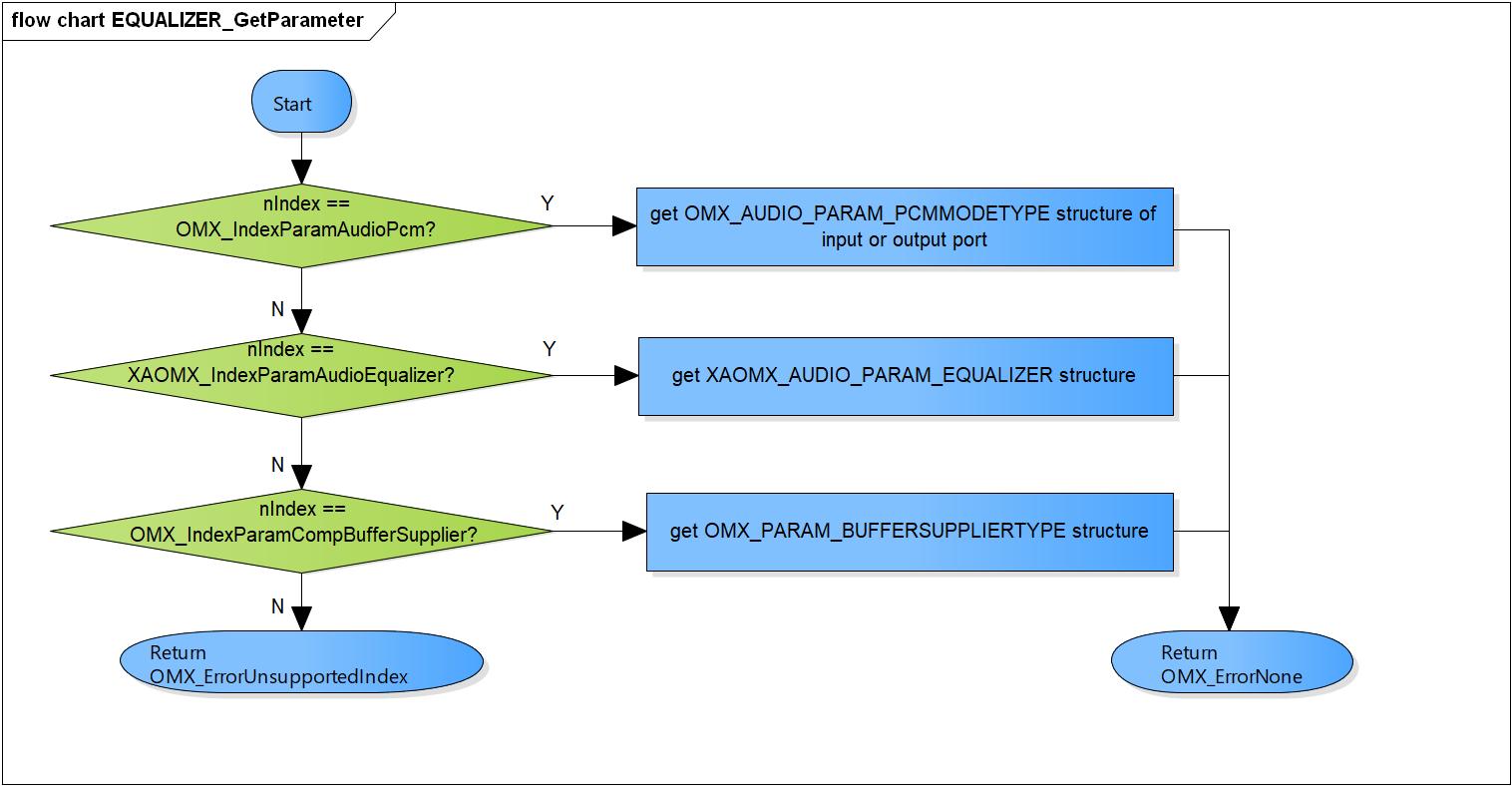


Figure 3‑5 EQUALIZER\_GetParameter flowchart

### EQUALIZER\_SetParameter

DD\_API\_EQZ\_01\_006

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static OMX\_ERRORTYPE EQUALIZER\_SetParameter(XAOMXCodecBase \*pBase,  OMX\_INDEXTYPE nIndex,  OMX\_PTR pParam) | | | |
| **Function** | Set parameter from pParam and save to parameter structure. | | | |
| **Arguments** | Type | Name | I/O | Description |
| XAOMXCodecBase\* | pBase | I/O | Pointer to base codec structure of Equalizer component. |
| OMX\_INDEXTYPE | nIndex | I | Index of parameter structure.  Supported range:  OMX\_IndexParamAudioPcm  XAOMX\_IndexParamAudioEqualizer  OMX\_IndexParamCompBufferSupplier |
| OMX\_PTR | pParam | O | Pointer to parameter structure want to set. |
| **Return value** | OMX\_ErrorNone | | Normal return. | |
| OMX\_ErrorUnsupportedIndex | | Wrong Index of structure. | |
| OMX\_ErrorBadPortIndex | | Port index is invalid | |
| **Description** | * EQUALIZER\_SetParameter processing:   - Set parameters of OMX\_AUDIO\_PARAM\_PCMMODETYPE, XAOMX\_AUDIO\_PARAM\_EQUALIZER, OMX\_PARAM\_BUFFERSUPPLIERTYPE structure base on nIndex. | | | |

[Covers: FD\_API\_EQZ\_005]

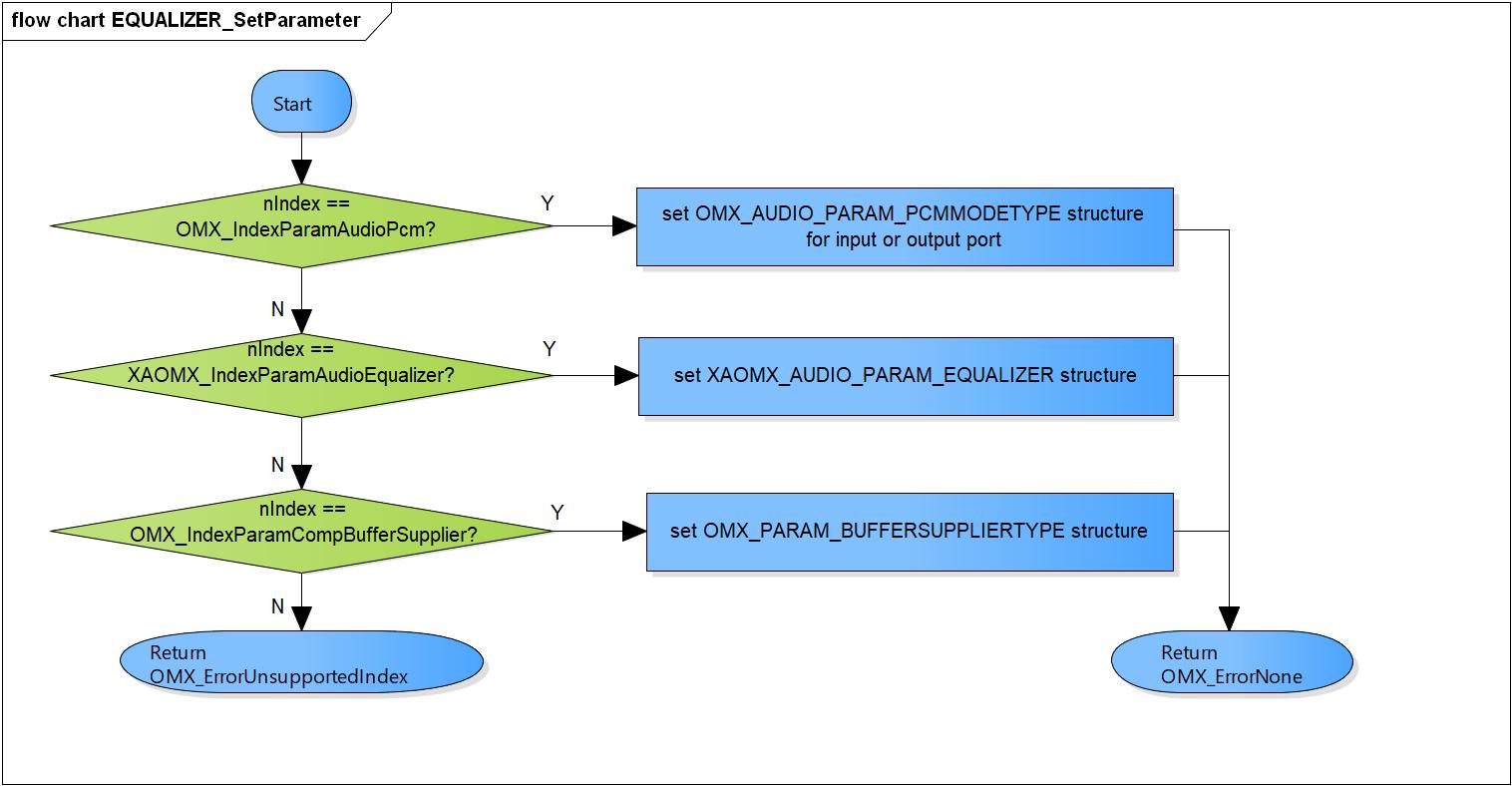


Figure 3‑6 EQUALIZER\_SetParameter flowchart

### EQUALIZER\_ComponentInit

DD\_API\_EQZ\_01\_007

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static OMX\_ERRORTYPE EQUALIZER\_ComponentInit(xf\_proxy\_t \*proxy,  OMX\_HANDLETYPE hComponent,  OMX\_PTR pAppData,  OMX\_CALLBACKTYPE \*pCallbacks) | | | |
| **Function** | Initialize base codec interface, parameters, PCM and Equalizer format. | | | |
| **Arguments** | Type | Name | I/O | Description |
| xf\_proxy\_t\* | proxy | O | OpenMAX interface proxy. |
| OMX\_HANDLETYPE | hComponent | O | Pointer to Equalizer component handle. |
| OMX\_PTR | pAppData | I | Pointer to pAppData of Equalizer component. |
| OMX\_CALLBACKTYPE | pCallbacks | I | Pointer to Callbacks of Equalizer component. |
| **Return value** | OMX\_ErrorNone | | Normal return. | |
| OMX\_ErrorInsufficientResources | | Initialize base codec interface is fail. | |
| **Description** | * EQUALIZER\_ComponentInit processing:   - Initialize base codec interface, parameters, PCM and Equalizer format. | | | |

[Covers: FD\_API\_EQZ\_001]

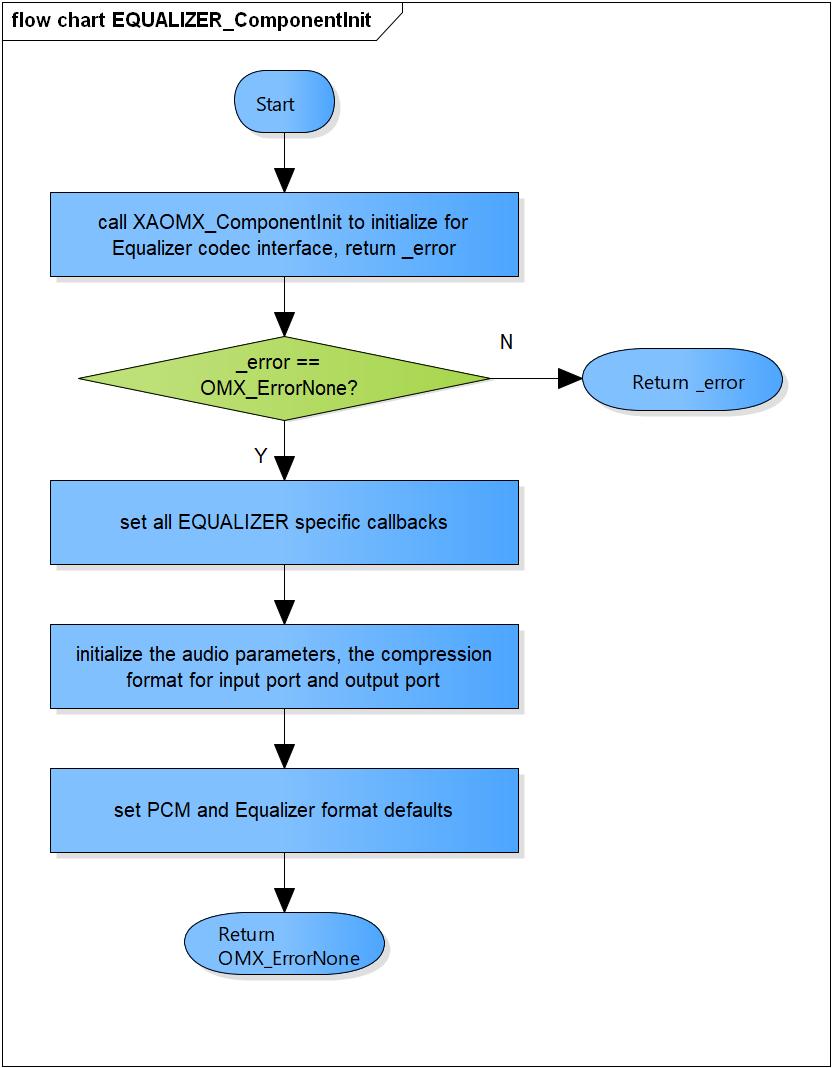


Figure 3‑7 EQUALIZER\_ComponentInit flowchart

### EQUALIZER\_ComponentDeInit

DD\_API\_EQZ\_01\_008

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | static OMX\_ERRORTYPE EQUALIZER\_ComponentDeInit(OMX\_HANDLETYPE hComponent) | | | |
| **Function** | Destroy Equalizer component and private data. | | | |
| **Arguments** | Type | Name | I/O | Description |
| OMX\_HANDLETYPE | hComponent | O | Pointer to Equalizer component handle. |
| **Return value** | OMX\_ErrorNone | | Normal return. | |
| OMX\_ErrorBadParameter | | Private data is NULL. | |
| **Description** | * EQUALIZER\_ComponentDeInit processing:   - Destroy Equalizer component and private data. | | | |

[Covers: FD\_API\_EQZ\_002]

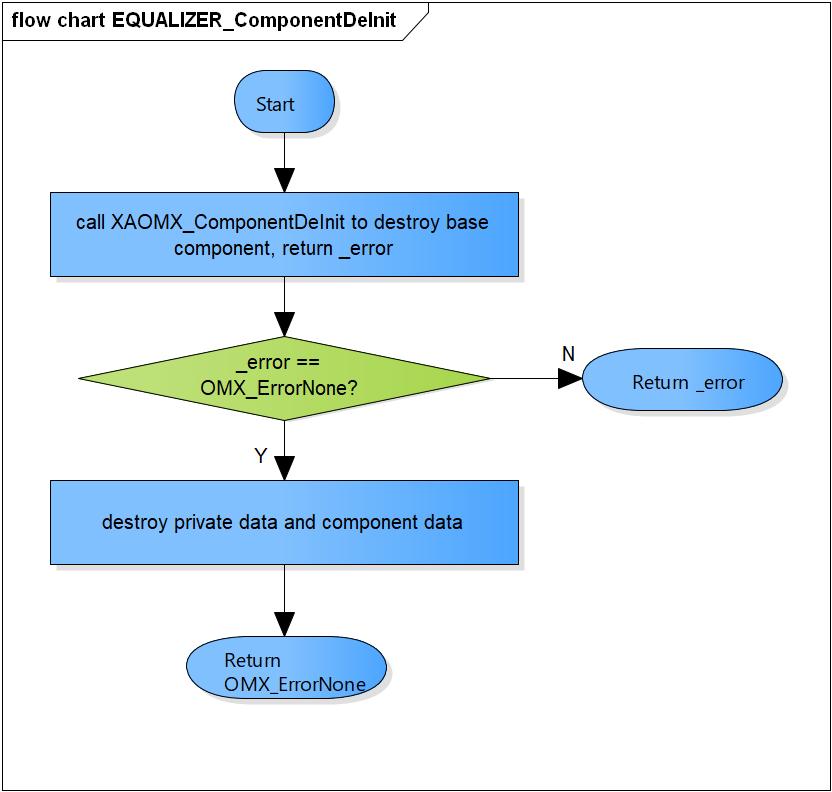


Figure 3‑8 EQUALIZER\_ComponentDeInit flowchart

### EQUALIZER\_ComponentCreate

DD\_API\_EQZ\_01\_009

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Syntax** | OMX\_ERRORTYPE EQUALIZER\_ComponentCreate(xf\_proxy\_t \*proxy, OMX\_HANDLETYPE \*hComponent, OMX\_PTR pAppData, OMX\_CALLBACKTYPE \*pCallbacks) | | | |
| **Function** | Create base component and initialize the Equalizer component. | | | |
| **Arguments** | Type | Name | I/O | Description |
| xf\_proxy\_t\* | proxy | O | OpenMAX interface proxy |
| OMX\_HANDLETYPE\* | hComponent | O | Pointer to Equalizer component handle |
| OMX\_PTR | pAppData | I | Pointer to pAppData of Equalizer component |
| OMX\_CALLBACKTYPE\* | pCallbacks | I | Pointer to Callbacks of Equalizer component |
| **Return value** | OMX\_ErrorNone | | Normal return. | |
| OMX\_ErrorInsufficientResources | | Cannot create the base component.  Cannot allocate private memory.  Initialize the Equalizer component is fail. | |
| **Description** | * EQUALIZER\_ComponentCreate processing:   - Create the base component.  - Allocate the private memory.  - Initialize the Equalizer component. | | | |

[Covers: FD\_API\_EQZ\_001]

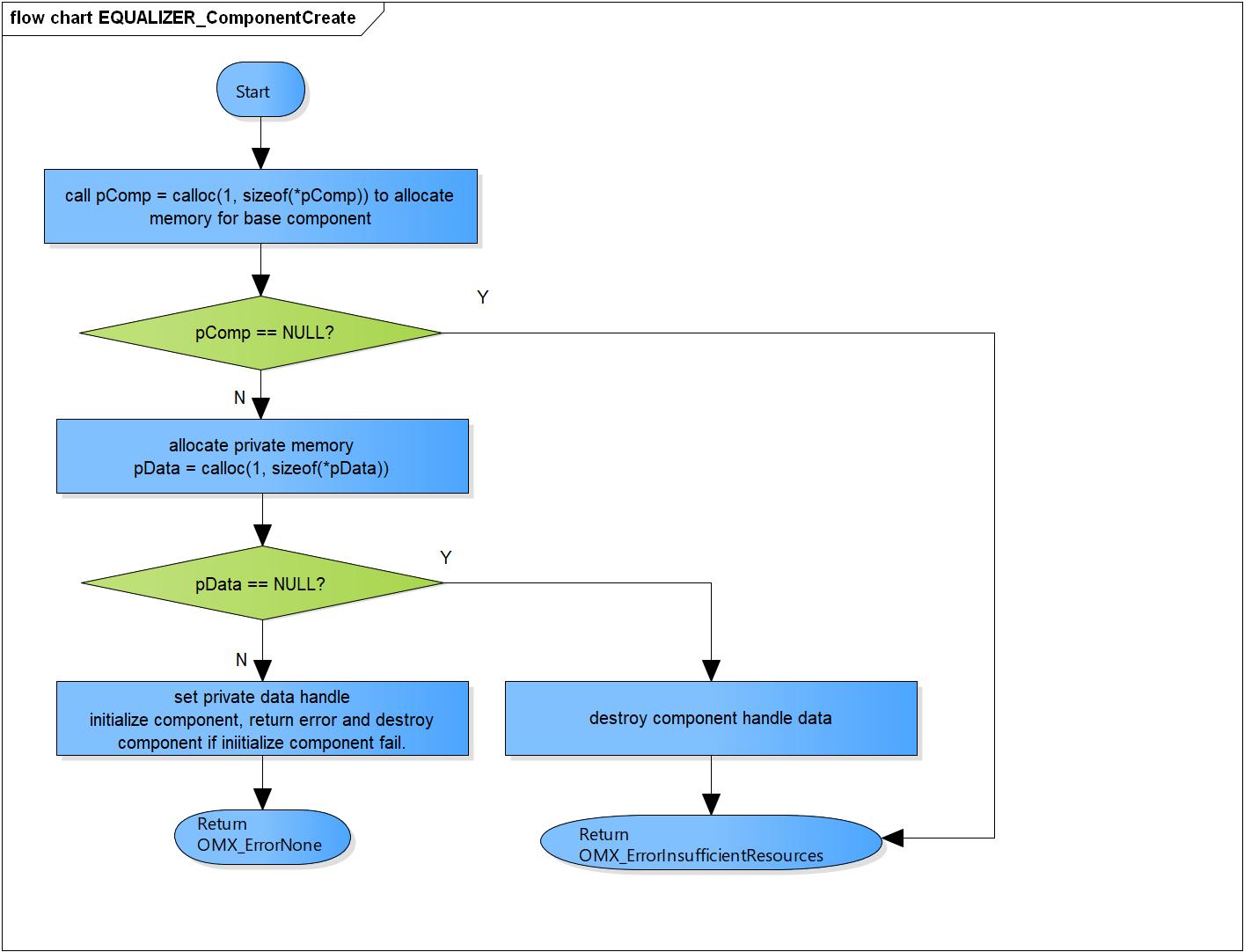


Figure 3‑9 EQUALIZER\_ComponentCreate flowchart

# 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 |