# Bang & Olufsen - ASE Protocol Buffers definition for FEP interface

Generated by Doxygen 1.8.11

# Contents

# 1 Introduction

This is the Protocol Buffers message definitions for the FEP interface on the ASE platform. All messages are prefixed with either "FepAse" or "AseFep" which indicates the direction. A message prefixed with "FepAse" will be sent from the FEP to the ASE and vice versa. Messages are always encapsulated in either FepAseMessage or AseFepMessage which contains the message ID of the embodied message.

Messages must be sent over the UART using the Yet Another HDLC (yahdlc) implementation located here: https://github.com/bang-olufsen/yahdlc

The acknowledgement of DATA frames must be done as soon as the integrity of the message has been checked to have the highest possible throughput. New requests, events and commands may be sent after the message acknowledgement. The sequence number must be used to handle retransmission and duplicated messages.

# 2 Module Index

## 2.1 Modules

Here is a list of all modules:



# 3 Class Index

## 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

# 4 File Index

## 4.1 File List

Here is a list of all documented files with brief descriptions:



# 5 Module Documentation

## 5.1 Volume

The speakers volume level is control by ASE, if volume is changed a message with the new volume level is sent to FEP.

### Classes

- struct AseFepVolumeChangedEvent

  *Event from ASE to FEP with new volume value, the speakers output volume is set to this value.*
- struct FepAseAbsoluteVolumeCommand

  *Command from FEP to ASE to change volume to a specific level, the requested volume change does not take effect until AseFepVolumeChangedEvent is received by FEP.*
- struct FepAseRelativeVolumeCommand

  *Command from FEP to ASE to change the volume relative to the current volume level, the requested volume change does not take effect until AseFepVolumeChangedEvent is received by FEP.*
- struct AseFepVolumeFadeEvent

  *Perform a continuous volume change from one volume level to another volume level in a defined time frame.*
- struct FepAseVolumeFadeComplete

  *Continuous volume change operation has completed.*

### 5.1.1 Detailed Description

The speakers volume level is control by ASE, if volume is changed a message with the new volume level is sent to FEP.

The ASE can receive requests to change volume from many different sources such as Bluetooth remote controller, Airplay, NetworkLink, local user interface (FEP) etc. When a request to change volume is received, various optional volume level limitations are enforced and the resulting volume level is forwarded to the DSP via FEP.

## 5.2 Audio PCM formats

Command from ASE to FEP to change the bit clock frequency of the I2S bus when the audio format changes, e.g.

#### Classes

- struct AseFepSetAudioPcmFormatCommand
- struct FepAseSetAudioPcmFormatCommandResp

  *Response from FEP when setting PCM format.*

### 5.2.1 Detailed Description

Command from ASE to FEP to change the bit clock frequency of the I2S bus when the audio format changes, e.g.

if the sample rate changes from 44.1 kHz to 48 kHz.

## 5.3 Communication Watchdog

To detect and recover from malfunction, the interface between ASE and FEP is monitored for communication activity and errors.

### Classes

- struct FepAseAliveReq

  *Request the ASE to give proof of life.*
- struct AseFepAliveResp

  *Alive message returned to FEP on request.*

### 5.3.1 Detailed Description

To detect and recover from malfunction, the interface between ASE and FEP is monitored for communication activity and errors.

### ASE reset supervision

When the ASE is reset a timer is started in FEP to supervise that ASE comes back online. If ASE does not start to communicate with FEP within a given timeout, FEP must reset ASE to give it another try.

In the normal startup situation the timeout is 2 minutes but in software update state the timeout is 15 minutes, because ASE is unresponsive during parts of the software update process and should not be interrupted.

In ASE a restart counter is maintained to track the number of restarts without reaching the operational state. If the restart counter exceeds 3, ASE will revert all settings to factory default and attempt to restart again.

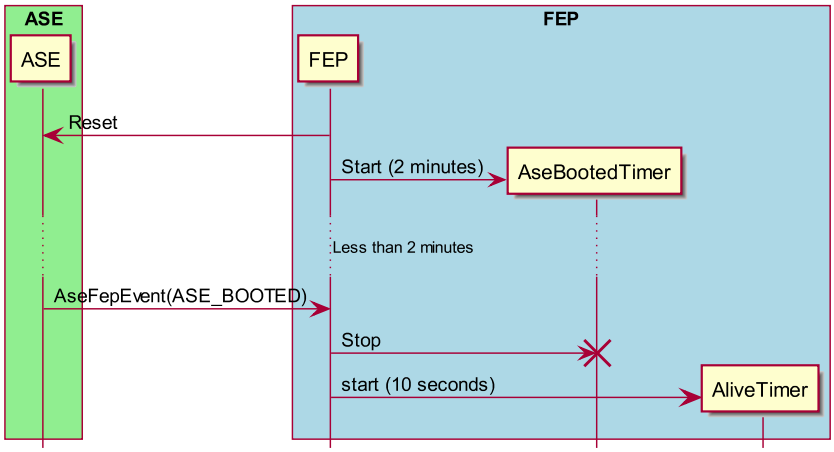


**Figure 1 ; Normal startup**

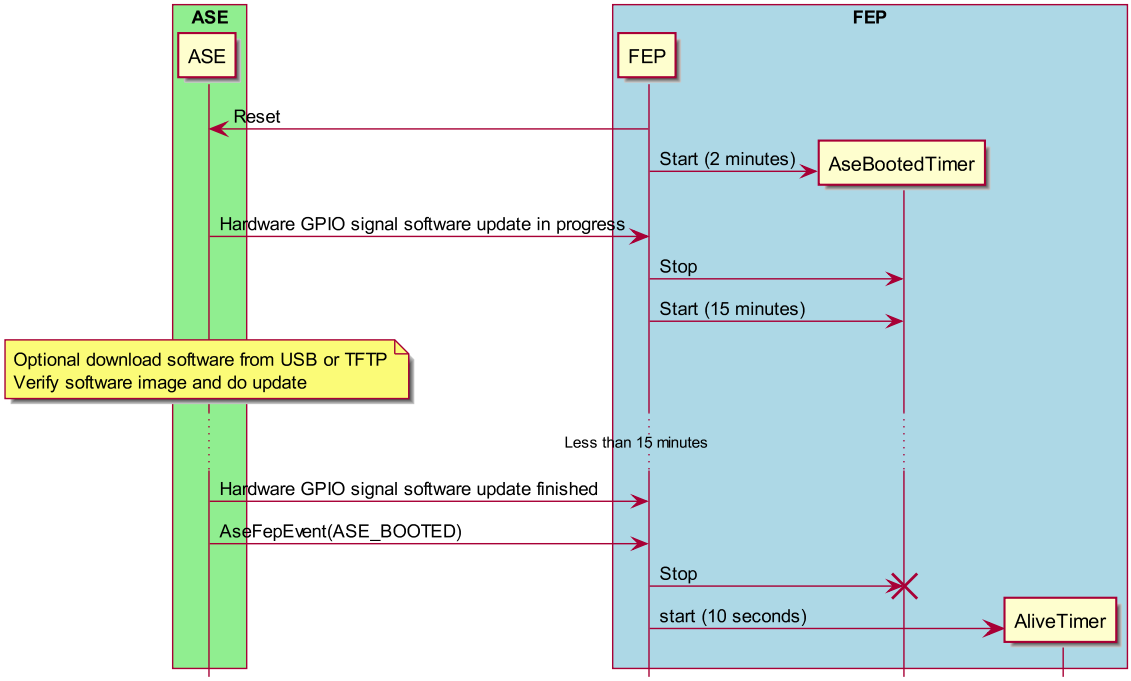


**Figure 2 ; Software update**

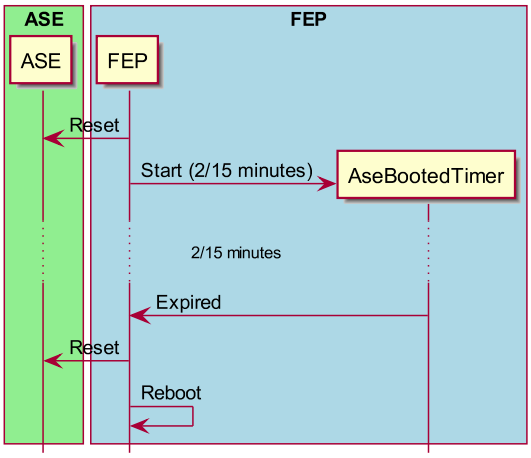


**Figure 3 ; ASE failed to boot**

### Alive supervision

During normal operation the communication between ASE and FEP is monitored, and if communication stops the system is restarted in the attempt to restore communication. The survailance relys on both "normal" communication and the special FepAseAliveReq and AseFepAliveResp messages. If no normal communication has been detected for 10 seconds, FEP sends the FepAseAliveReq to verify that ASE is still alive. After 3 consecutive alive requests without a response, FEP deems ASE unable to recover by itself and restarts the system.

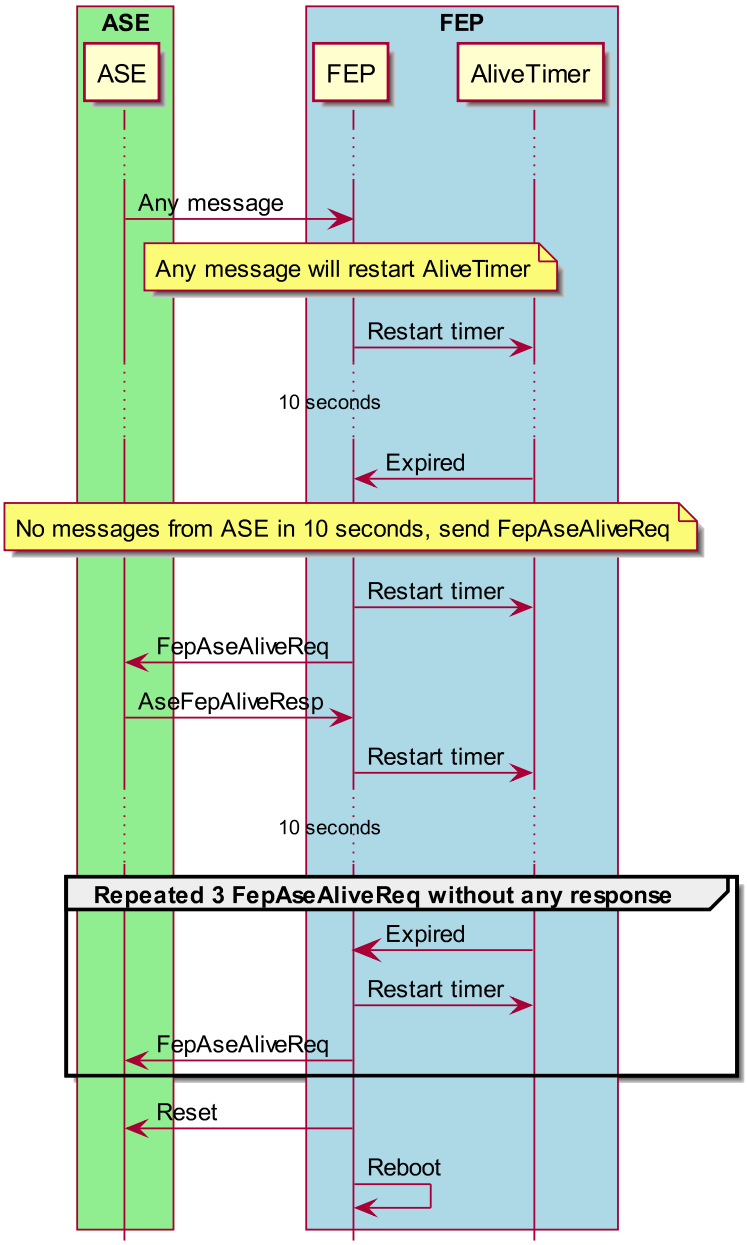


**Figure 4 ; No response to alive requests**

## 5.4 Logging and statistics

Write an entry in the product log.

### Classes

- struct FepAseProductLog
- struct FepAseServiceLog

  *Write an entry in the service log.*

### 5.4.1 Detailed Description

Write an entry in the product log.

## 5.5 Position sound mode

Position sound mode setting.

### Classes

- struct PositionSoundMode
- struct AseFepSetPositionSoundModeReq

  *Sets the position sound mode.*
- struct AseFepGetPositionSoundModeReq

  *Gets the current selected position sound mode.*
- struct FepAsePositionSoundModeResp

  *Response when setting position sound mode.*

### 5.5.1 Detailed Description

Position sound mode setting.

## 5.6 Speaker Calibration

Internal speaker.

### Classes

- struct InternalSpeaker
- struct AseFepSetInternalSpeakerCompensationCommand

  *Set the internal speaker compensation gain for one or more speaker units.*
- struct FepAseInternalSpeakerCompensationResp

  *Response when setting or getting internal speaker compensation.*

### 5.6.1 Detailed Description

Internal speaker.

## 5.7 Network status

On request from FEP or when status changes on one or more network interfaces ASE will send AseFepNetworkInfo to FEP.

### Classes

- struct WiFi
- struct NetworkInfo
- struct AseFepNetworkInfo

  *Current network status information. Sent when network status has changed or upon request from FEP.*

### 5.7.1 Detailed Description

On request from FEP or when status changes on one or more network interfaces ASE will send AseFepNetworkInfo to FEP.

## 5.8 Message ID

To identify the message being received, a message ID is prepended to every message.

#### Classes

- struct AseFepMessage
- struct FepAseMessage

### 5.8.1 Detailed Description

To identify the message being received, a message ID is prepended to every message.

# 6 Class Documentation

## 6.1 AseFepAliveResp Struct Reference

Alive message returned to FEP on request.

### 6.1.1 Detailed Description

Alive message returned to FEP on request.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.2 AseFepEvent Struct Reference

Event from ASE to FEP.

#### Public Types

- enum Event {
  SW_UPDATE_STARTED = 0, **SW_UPDATE_FINISHED** = 1, **SW_UPDATE_FAILED** = 2, **BT_PAIRING_ENABLED** = 3,
  **BT_PAIRING_DISABLED** = 4, **BT_PAIRING_FAILED** = 5, **BT_PAIRING_SUCCEEDED** = 6, **BTLE_PAIRING_ENABLED** = 7,
  **BTLE_PAIRING_DISABLED** = 8, **BTLE_PAIRING_FAILED** = 9, **BTLE_PAIRING_SUCCEEDED** = 10, **PLAYER_PLAYING** = 11,
  **PLAYER_PAUSED** = 12, **PLAYER_STOPPED** = 13, **PLAYER_MUTED** = 14, **PLAYER_UNMUTED** = 15,
  **SOFTAP_STARTED** = 16, **SOFTAP_STOPPED** = 17, **APPLE_WAC_STARTED** = 20, **APPLE_WAC_STOPPED** = 21,
  **APPLE_WAC_TIMEOUT** = 22, ASE_BOOTED = 23, ASE_OVERHEAT_ALERT = 24, SW_UPDATE_FEP_BOOTLOADER = 25,
  SW_UPDATE_FEP_APPLICATION = 26, SYSTEM_STATUS_STORAGE = 27, SYSTEM_STATUS_STANDBY = 28, SYSTEM_STATUS_ON = 29,
  SYSTEM_STATUS_OFF = 30, FACTORY_RESET_START = 31, FACTORY_RESET_DONE = 32, SYSTEM_STATUS_ASE_RESTART = 33,
  SYSTEM_STATUS_RESTART = 34, SYSTEM_STATUS_ON_NO_OPERATION = 35, COMFORT_TONE_START = 36, COMFORT_TONE_DONE = 37 }

#### Public Attributes

- optional Event **event** = 1

### 6.2.1 Detailed Description

Event from ASE to FEP.

#### 6.2.2 Member Enumeration Documentation

##### 6.2.2.1 enum AseFepEvent::Event

**Enumerator**

***SW_UPDATE_STARTED*** Software update of the ASE has started, local UI must show software update in progress. ASE reboot is part of the update process, if new firmware is available for the FEP, firmware update starts after ASE reboot.

***ASE_BOOTED*** ASE has completed the boot sequence and is ready to receive messages from FEP.

***ASE_OVERHEAT_ALERT*** ASE is overheated shut down ASE to prevent permanent damage.

***SW_UPDATE_FEP_BOOTLOADER*** FEP must prepare for FEP bootloader update.

***SW_UPDATE_FEP_APPLICATION*** FEP must prepare for FEP application update.

***SYSTEM_STATUS_STORAGE*** ASE is ready for storage mode (to be put on storage shelves)

***SYSTEM_STATUS_STANDBY*** Event send when ASE enter source standby.

***SYSTEM_STATUS_ON*** Event send when exiting standby (a source has started from standby)

***SYSTEM_STATUS_OFF*** Event send when ASE is ready to be powered off.

***FACTORY_RESET_START*** Event send when >>reset to factory<< have been initiated.

***FACTORY_RESET_DONE*** Event send when >>reset to factory<< is done.

***SYSTEM_STATUS_ASE_RESTART*** ASE is going to restart, FEP will gracefully shut down DSP and amplifier and signal reboot on the local UI.

***SYSTEM_STATUS_RESTART*** Request to restart the entire product. FEP will gracefully shut down DSP and amplifier, signal reboot on the local UI and reboot the product.

***SYSTEM_STATUS_ON_NO_OPERATION*** Event send when the product is ON, but no user interaction is registered for a given time period, default is 5 minutes. How the FEP will react to the event is product specific, dimming the LEDs is one option.

***COMFORT_TONE_START*** ASE indicates to FEP when the comfort sound playback starts. FEP has to turn on the DSP in case the DSP is off.

***COMFORT_TONE_DONE*** ASE indicates to FEP when the comfort sound playback is finished. FEP can turn off the DSP in case it was off before the comfort sound playback.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

### 6.3 AseFepGetPositionSoundModeReq Struct Reference

Gets the current selected position sound mode.

#### 6.3.1 Detailed Description

Gets the current selected position sound mode.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.4 AseFepMessage Struct Reference

### Public Attributes

- oneof **OneOf**
- AseFepVolumeChangedEvent **aseFepVolumeChangedEvent** = 3
- AseFepSetLineInSensitivity **aseFepSetLineInSensitivity** = 4
- AseFepTunnel **aseFepTunnel** = 6
- AseFepSetAudioInputReq **aseFepSetAudioInputReq** = 7
- AseFepSetAudioOutputReq **aseFepSetAudioOutputReq** = 8
- AseFepAliveResp **aseFepAliveResp** = 9
- AseFepSetPositionSoundModeReq **aseFepSetPositionSoundModeReq** = 11
- AseFepGetPositionSoundModeReq **aseFepGetPositionSoundModeReq** = 12
- AseFepNetworkInfo **aseFepNetworkInfo** = 13
- AseFepReq **aseFepReq** = 14
- AseFepSetAudioPcmFormatCommand **aseFepSetAudioPcmFormatCommand** = 15
- AseFepSetInternalSpeakerCompensationCommand **aseFepSetInternalSpeakerCompensationCommand** = 16
- AseFepVolumeFadeEvent **aseFepVolumeFadeEvent** = 17

### 6.4.1 Member Data Documentation

#### 6.4.1.1 oneof AseFepMessage::OneOf

**Initial value:**

```
{
    AseFepEvent aseFepEvent = 2
```

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.5 AseFepNetworkInfo Struct Reference

Current network status information. Sent when network status has changed or upon request from FEP.

### Public Attributes

- repeated NetworkInfo **networkInfo** = 1

### 6.5.1 Detailed Description

Current network status information. Sent when network status has changed or upon request from FEP.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.6 AseFepReq Struct Reference

Send request from ASE to FEP.

### Public Types

- enum Request {
  PRODUCT_INFO = 0, POWER_STATUS = 1, VERSION_INFO = 2, STATISTICS = 3,
  NETWORK_INFO = 4, INTERNAL_SPEAKER_COMPENSATION = 5 }

### Public Attributes

- required Request **request** = 1

#### 6.6.1 Detailed Description

Send request from ASE to FEP.

#### 6.6.2 Member Enumeration Documentation

##### 6.6.2.1 enum AseFepReq::Request

**Enumerator**

***PRODUCT_INFO*** Request the product info from FEP, reply is FepAseProductInfo.

***POWER_STATUS*** Request power status, reply is FepAsePowerStatus.

***VERSION_INFO*** Request version info from FEP, reply is FepAseVersionInfo.

***STATISTICS*** Request counters, timers and other statistical information reply is FepAseStatistics.

***NETWORK_INFO*** Request network information, reply is AseFepNetworkInfo.

***INTERNAL_SPEAKER_COMPENSATION*** Get the current internal speaker compensation gain for all speaker units. FEP replies with FepAseInternalSpeakerCompensationResp.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.7 AseFepSetAudioInputReq Struct Reference

Request to set audio input.

### Public Types

- enum **AudioInput** {
  **ASE** = 0, **LINE** = 1, **TOS_LINK** = 2, **POWER_LINK** = 3,
  **HDMI** = 4, **WIRELESS_MULTICHANNEL** = 5 }

#### Public Attributes

- optional AudioInput input = 1

  *Configure the audio input path.*
- optional bool local = 2

  *Configure whether to use local playback or not when ever the input/output path allows it.*

### 6.7.1 Detailed Description

Request to set audio input.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.8 AseFepSetAudioOutputReq Struct Reference

Request to set audio output.

#### Public Types

- enum **AudioOutput** {
  **AMPLIFIER** = 0, **LINE** = 1, **TOS_LINK** = 2, **POWER_LINK** = 3,
  **HDMI** = 4, **WIRELESS_MULTICHANNEL** = 5 }

#### Public Attributes

- repeated AudioOutput output = 1 [packed=true]

  *Select one or more audio output paths.*

### 6.8.1 Detailed Description

Request to set audio output.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.9 AseFepSetAudioPcmFormatCommand Struct Reference

### Public Types

- enum Format {
  F32BE = 0, F32LE = 1, F64BE = 2, F64LE = 3,
  S16BE = 4, S16LE = 5, S24BE = 6, S24LE = 7,
  S32BE = 8, S32LE = 9, S8 = 10, U16BE = 11,
  U16LE = 12, U24BE = 13, U24LE = 14, U32BE = 15,
  U32LE = 16, U8 = 17 }

  *The format of each sample.*
- enum Rate {
  RATE_8_KHZ = 0, RATE_12_KHZ = 1, RATE_16_KHZ = 2, RATE_24_KHZ = 3,
  RATE_32_KHZ = 4, RATE_48_KHZ = 5, RATE_64_KHZ = 6, RATE_96_KHZ = 7,
  RATE_11025_HZ = 10, RATE_22050_HZ = 11, RATE_44100_HZ = 12, RATE_88200_HZ = 13 }

  *The sampling rate.*
- enum Channels { **MONO** = 0, **STEREO** = 1 }

  *The channels in the audio.*

### Public Attributes

- required Format **format** = 1
- required Rate **sampling** = 2
- required Channels **channels** = 3

### 6.9.1 Member Enumeration Documentation

#### 6.9.1.1 enum AseFepSetAudioPcmFormatCommand::Format

The format of each sample.

#### Enumerator

***F32BE*** 32-bit floating-point big-endian

***F32LE*** 32-bit floating-point little-endian

***F64BE*** 64-bit floating-point big-endian

***F64LE*** 64-bit floating-point little-endian

***S16BE*** signed 16-bit big-endian

***S16LE*** signed 16-bit little-endian

***S24BE*** signed 24-bit big-endian

***S24LE*** signed 24-bit little-endian

***S32BE*** signed 32-bit big-endian

***S32LE*** signed 32-bit little-endian

***S8*** signed 8-bit

***U16BE*** unsigned 16-bit big-endian

***U16LE*** unsigned 16-bit little-endian

***U24BE*** unsigned 24-bit big-endian

***U24LE*** unsigned 24-bit little-endian

***U32BE*** unsigned 32-bit big-endian

***U32LE*** unsigned 32-bit little-endian

***U8*** unsigned 8-bit

#### 6.9.1.2 enum AseFepSetAudioPcmFormatCommand::Rate

The sampling rate.

**Enumerator**

***RATE_8_KHZ*** 8 kHz
***RATE_12_KHZ*** 12 kHz
***RATE_16_KHZ*** 16 kHz
***RATE_24_KHZ*** 24 kHz
***RATE_32_KHZ*** 32 kHz
***RATE_48_KHZ*** 48 kHz
***RATE_64_KHZ*** 64 kHz
***RATE_96_KHZ*** 96 kHz
***RATE_11025_HZ*** 11.025 kHz
***RATE_22050_HZ*** 22.05 kHz
***RATE_44100_HZ*** 44.1 kHz
***RATE_88200_HZ*** 88.2 kHz

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.10 AseFepSetInternalSpeakerCompensationCommand Struct Reference

Set the internal speaker compensation gain for one or more speaker units.

**Public Attributes**

- repeated InternalSpeaker **internalSpeaker** = 1

### 6.10.1 Detailed Description

Set the internal speaker compensation gain for one or more speaker units.

If a speaker unit's performance deviates from the specifications, a gain value can be applied to compensated for the deviation. The FEP or one of its sub-components persist the compensation gain in non-volatile memory. FEP replies with FepAseInternalSpeakerCompensationResp.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.11 AseFepSetLineInSensitivity Struct Reference

Set the line-in sensitivity level.

#### Public Types

- enum LineInSensitivity { HIGH = 0, MEDIUM = 1, LOW = 2, DISABLED = 3 }

#### Public Attributes

- optional LineInSensitivity **sensitivity** = 1

### 6.11.1 Detailed Description

Set the line-in sensitivity level.

### 6.11.2 Member Enumeration Documentation

#### 6.11.2.1 enum AseFepSetLineInSensitivity::LineInSensitivity

#### Enumerator

***HIGH*** Line-in is triggered by a weak input signal. Suitable for most MP3 players.

***MEDIUM*** Line-in is triggered by a medium input signal. Use with standard audio equipment and computers.

***LOW*** Line-in is triggered by a strong input signal. Suitable for DVD/BD players.

***DISABLED*** Line-in sense is disabled.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.12 AseFepSetPositionSoundModeReq Struct Reference

Sets the position sound mode.

#### Public Attributes

- optional PositionSoundMode **mode** = 1

### 6.12.1 Detailed Description

Sets the position sound mode.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.13 AseFepTunnel Struct Reference

Message intended for debugging and production test.

### Public Attributes

- optional bytes **data** = 1

### 6.13.1 Detailed Description

Message intended for debugging and production test.

The data included in the message can have any arbitrary format agreed between production test FEP software development to support hardware verification on the production line.

Typically the message is used to request information or exercise a specific hardware component.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.14 AseFepVolumeChangedEvent Struct Reference

Event from ASE to FEP with new volume value, the speakers output volume is set to this value.

### Public Attributes

- optional uint32 volume = 1

  *FEP must adjust the speaker volume to this value.*

### 6.14.1 Detailed Description

Event from ASE to FEP with new volume value, the speakers output volume is set to this value.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.15 AseFepVolumeFadeEvent Struct Reference

Perform a continuous volume change from one volume level to another volume level in a defined time frame.

#### Public Attributes

- optional uint32 start_volume = 1

  *Fade volume starting from start_volume. If omitted the fade operation starts from current volume level.*
- required uint32 target_volume = 2

  *Fade volume to target_volume.*
- required uint32 fade_duration = 3

  *In milliseconds the duration of the fade operation.*

### 6.15.1 Detailed Description

Perform a continuous volume change from one volume level to another volume level in a defined time frame.

During the fading, ASE discards any other volume change originated by any means (touch, remote, BNR, etc), therefore FEP must send FepAseVolumeFadeComplete to ASE when fading is complete.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.16 FepAseAbsoluteVolumeCommand Struct Reference

Command from FEP to ASE to change volume to a specific level, the requested volume change does not take effect until AseFepVolumeChangedEvent is received by FEP.

#### Public Attributes

- optional uint32 **volume** = 1

### 6.16.1 Detailed Description

Command from FEP to ASE to change volume to a specific level, the requested volume change does not take effect until AseFepVolumeChangedEvent is received by FEP.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.17 FepAseAliveReq Struct Reference

Request the ASE to give proof of life.

#### 6.17.1 Detailed Description

Request the ASE to give proof of life.

With an interval of one minute FEP requests ASE for proof of life to ensure that the communication channel is open and ASE is fully operational.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.18 FepAseAudioSocketChangeEvent Struct Reference

Event from FEP to ASE when a plug, e.g. mini-jack, have been inserted or removed from a (audio) socket.

#### Public Types

- enum AudioSocket { **MINI_JACK** = 0 }

  *The type of audio socket.*
- enum SocketState { INSERTED = 0, REMOVED = 1 }

#### Public Attributes

- optional AudioSocket **socket** = 1
- optional SocketState **state** = 2

#### 6.18.1 Detailed Description

Event from FEP to ASE when a plug, e.g. mini-jack, have been inserted or removed from a (audio) socket.

#### 6.18.2 Member Enumeration Documentation

##### 6.18.2.1 enum FepAseAudioSocketChangeEvent::SocketState

**Enumerator**

***INSERTED*** A plug has been inserted into the socket.

***REMOVED*** A plug has been removed from the socket.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.19 FepAseCommand Struct Reference

Command request from FEP to ASE (Local UI).

#### Public Types

- enum Command {
  JOIN = 0, **PLAY** = 1, **PAUSE** = 2, **STOP** = 3,
  **NEXT** = 4, **PREV** = 5, **BT_PAIRING_ON** = 6, **BT_PAIRING_OFF** = 7,
  **BTLE_PAIRING_ON** = 8, **BTLE_PAIRING_OFF** = 9, STANDBY = 10, OFF = 11,
  FACTORY_RESET = 12, **MUTE** = 13, SOUND = 14, SILENCE = 15,
  **NETWORK_SETUP** = 16, **SOUND_SILENCE_TOGGLE** = 17, **BT_PAIRING_TOGGLE** = 18, **PLAY_PAU**↩
  **SE_TOGGLE** = 19,
  **UNMUTE** = 20, **NEXT_SOURCE** = 21, STORAGE = 22, IOT = 23,
  ALL_STANDBY = 24 }

#### Public Attributes

- optional Command **command** = 1

### 6.19.1 Detailed Description

Command request from FEP to ASE (Local UI).

### 6.19.2 Member Enumeration Documentation

#### 6.19.2.1 enum FepAseCommand::Command

**Enumerator**

***JOIN*** One way join. Join/next joinable playback session.

***STANDBY*** Standby command, FEP will receive SYSTEM_STATUS_STANDBY when ASE have entered standby.

***OFF*** Off command, FEP will receive SYSTEM_STATUS_OFF when ASE have entered >>OFF<< mode.

***FACTORY_RESET*** Factory reset command, FEP will receive FACTORY_RESET_DONE when ASE have finished factory reset.

***SOUND*** The sound command has the functionality defined by ASE1.x. In short that is start playback from the top of the prioritized source list or skip to next source if already playing.

***SILENCE*** The silence command takes the necessary means to make the product silence i.e. pause if possible otherwise mute.

***STORAGE*** Prepare for storage i.e. gracefully shut down and send AseFepEvent(SYSTEM_STATUS_ST↩ ORAGE) when ready.

***IOT*** The IoT button, reserved for future use.

***ALL_STANDBY*** ALL STANDBY command.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.20 FepAseEvent Struct Reference

Event from FEP to ASE.

#### Public Types

- enum **Event** {
  **LINE_SENSE_ACTIVE** = 0, **LINE_SENSE_INACTIVE** = 1, **TOSLINK_SENSE_ACTIVE** = 2, **TOSLINK_S↩ ENSE_INACTIVE** = 3,
  **HDMI_SENSE_ACTIVE** = 4, **HDMI_SENSE_INACTIVE** = 5 }

#### Public Attributes

- optional Event **event** = 1

### 6.20.1 Detailed Description

Event from FEP to ASE.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.21 FepAseInternalSpeakerCompensationResp Struct Reference

Response when setting or getting internal speaker compensation.

#### Public Types

- enum **Error** {
  **NO_ERROR** = 1, **POSITION_ERROR** = 2, **TYPE_ERROR** = 3, **POSITION_TYPE_COMBINATION_ERROR** = 4,
  **GAIN_ERROR** = 5 }

#### Public Attributes

- required Error **error** = 1
- repeated InternalSpeaker internalSpeaker = 2

  *Current internal speaker compensation.*

### 6.21.1 Detailed Description

Response when setting or getting internal speaker compensation.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.22 FepAseMessage Struct Reference

### Public Attributes

- oneof **OneOf**
- FepAseEvent **fepAseEvent** = 3
- FepAseAbsoluteVolumeCommand **fepAseAbsoluteVolumeCommand** = 4
- FepAseRelativeVolumeCommand **fepAseRelativeVolumeCommand** = 5
- FepAseTunnel **fepAseTunnel** = 7
- FepAseSetAudioInputResp **fepAseSetAudioInputResp** = 8
- FepAseSetAudioOutputResp **fepAseSetAudioOutputResp** = 9
- FepAseAliveReq **fepAseAliveReq** = 10
- FepAseProductLog **fepAseProductLog** = 11
- FepAseServiceLog **fepAseServiceLog** = 12
- FepAsePositionSoundModeResp **fepAsePositionSoundModeResp** = 14
- FepAsePlayComfortToneCmd **fepAsePlayComfortToneCmd** = 16
- FepAseProductInfo **fepAseProductInfo** = 17
- FepAsePowerStatus **fepAsePowerStatus** = 18
- FepAseVersionInfo **fepAseVersionInfo** = 19
- FepAseStatistics **fepAseStatistics** = 20
- FepAseAudioSocketChangeEvent **fepAseAudioSocketChangeEvent** = 21
- FepAseSetAudioPcmFormatCommandResp **fepAseSetAudioPcmFormatCommandResp** = 22
- FepAseInternalSpeakerCompensationResp **fepAseInternalSpeakerCompensationResp** = 23
- FepAseVolumeFadeComplete **fepAseVolumeFadeComplete** = 24
- FepAseReq **fepAseReq** = 25

### 6.22.1 Member Data Documentation

#### 6.22.1.1 oneof FepAseMessage::OneOf

**Initial value:**

```
{
    FepAseCommand fepAseCommand = 2
```

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.23 FepAsePlayComfortToneCmd Struct Reference

Instruct ASE to play a comfort tone.

### Public Attributes

- optional string tone = 1

  *Name of the wav file to play.*

#### 6.23.1 Detailed Description

Instruct ASE to play a comfort tone.

Comfort tones are stored as wav files on ASE file system. To play a specific comfort tone, the FEP includes the wav file name in the comfort tone request command.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

### 6.24 FepAsePositionSoundModeResp Struct Reference

Response when setting position sound mode.

#### Public Types

- enum **Error** { **NO_ERROR** = 1, **POSITION_ERROR** = 2, **ORIENTATION_ERROR** = 3, **COMBINATION_E↩ RROR** = 4 }

#### Public Attributes

- optional Error **error** = 1
- optional PositionSoundMode mode = 2

  *Current position sound mode.*

#### 6.24.1 Detailed Description

Response when setting position sound mode.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

### 6.25 FepAsePowerStatus Struct Reference

Sent upon AseFepReq(POWER_STATUS) request and when power/battery status changes.

#### Public Types

- enum **ACLineStatus** { **UNPLUGGED** = 0, **PLUGGED** = 1 }
- enum BatteryStatus {
  NO_BATTERY = 0, **LEVEL_CRITICAL** = 1, **LEVEL_LOW** = 2, **LEVEL_MIDDLE** = 3,
  **LEVEL_HIGH** = 4 }
- enum **BatteryHealthStatus** { **UNKNOWN** = 0, **GOOD** = 1, **POOR** = 2 }

#### Public Attributes

- required ACLineStatus **acLineStatus** = 1
- required BatteryStatus **batteryStatus** = 2
- optional uint32 batteryLevel = 3

  *The battery level in percent, 0-100%.*
- required BatteryHealthStatus **healthStatus** = 4

### 6.25.1 Detailed Description

Sent upon AseFepReq(POWER_STATUS) request and when power/battery status changes.

### 6.25.2 Member Enumeration Documentation

#### 6.25.2.1 enum FepAsePowerStatus::BatteryStatus

**Enumerator**

***NO_BATTERY*** The product does not have a battery.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.26 FepAseProductInfo Struct Reference

Response from AseFepReq(PRODUCT_INFO)

#### Public Types

- enum **ProductId** {
  **UNKNOWN** = 0, **FS1** = 1, **FS2** = 2, **SP3** = 3,
  **CA16** = 4 }

#### Public Attributes

- required string productName = 1

  *Product name used as default friendly name prefix.*
- required ProductId **productId** = 2

### 6.26.1 Detailed Description

Response from AseFepReq(PRODUCT_INFO)

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.27 FepAseProductLog Struct Reference

### Public Types

- enum Severity {
  DEBUG = 0, INFO = 1, NOTICE = 2, WARNING = 3,
  ERROR = 4 }

  *Severity of the incident causing the log entry to be written to the product log.*

### Public Attributes

- optional Severity severity = 1

  *Severity of the incident causing the log entry to be written.*
- optional string text = 2

  *Human readable text describing the incident.*
- optional string file = 3

  *File name of the source code file where the log entry is generated.*
- optional uint32 line = 4

  *Line number in the source code file where the log entry is generated.*

### 6.27.1 Member Enumeration Documentation

#### 6.27.1.1 enum FepAseProductLog::Severity

Severity of the incident causing the log entry to be written to the product log.

#### Enumerator

***DEBUG*** Log debugging statements that are not part of the release build.

***INFO*** Log internal tracing inside components.

***NOTICE*** Log internal tracing between components.

***WARNING*** Log additional information about warnings.

***ERROR*** Log additional information about error conditions.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.28 FepAseRelativeVolumeCommand Struct Reference

Command from FEP to ASE to change the volume relative to the current volume level, the requested volume change does not take effect until AseFepVolumeChangedEvent is received by FEP.

### Public Attributes

- optional int32 **volume** = 1

#### 6.28.1 Detailed Description

Command from FEP to ASE to change the volume relative to the current volume level, the requested volume change does not take effect until AseFepVolumeChangedEvent is received by FEP.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

### 6.29 FepAseReq Struct Reference

Request from FEP to ASE.

#### Public Types

- enum Request { GET_INTERNAL_SPEAKER_COMPENSATION = 0 }

#### Public Attributes

- required Request **request** = 1

#### 6.29.1 Detailed Description

Request from FEP to ASE.

#### 6.29.2 Member Enumeration Documentation

##### 6.29.2.1 enum FepAseReq::Request

**Enumerator**

***GET_INTERNAL_SPEAKER_COMPENSATION*** Get internal speaker compensation values, ASE replies with AseFepSetInternalSpeakerCompensationCommand.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

### 6.30 FepAseServiceLog Struct Reference

Write an entry in the service log.

#### Public Types

- enum Severity {
  NOTICE = 0, WARNING = 1, ERROR = 2, CRITICAL = 3,
  ALERT = 4, EMERGENCY = 5 }

#### Public Attributes

- optional Severity severity = 1

  *Severity of the incident causing the log entry to be written.*
- optional string text = 2

  *Human readable text describing the incident.*
- optional string file = 3

  *File name of the source code file where the log entry is generated.*
- optional uint32 line = 4

  *Line number in the source code file where the log entry is generated.*
- optional uint32 code = 5

  *Reserved for future use. Code identifying functional area causing the log entry.*

### 6.30.1 Detailed Description

Write an entry in the service log.

### 6.30.2 Member Enumeration Documentation

#### 6.30.2.1 enum FepAseServiceLog::Severity

**Enumerator**

***NOTICE*** Log a user action or external input. The audience of this logging is the service supporter or developer.

***WARNING*** Log a warning about a temporary error. The audience of this logging is the service supporter or developer.

***ERROR*** Log an error. The audience of this logging is the service technicians, so the logging message must be understandable and preferably suggest a possible solution.

***CRITICAL*** Log a critical error that prevents the user from performing a valid activity. The audience of this logging is the service technicians, so the logging message must be understandable and preferably suggest a possible solution.

***ALERT*** Log an error that requires immediate action. The audience of this logging is the service technicians, so the logging message must be understandable and preferably suggest a possible solution.

***EMERGENCY*** Log an error that the product has become unusable and must be replaced. The audience of this logging is the service technicians, so the logging message must be understandable and preferably suggest a possible solution.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.31 FepAseSetAudioInputResp Struct Reference

Response to a set audio input request.

#### Public Attributes

- optional bool **success** = 1

#### 6.31.1 Detailed Description

Response to a set audio input request.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

### 6.32 FepAseSetAudioOutputResp Struct Reference

Response to a set audio output request.

#### Public Attributes

- optional bool **success** = 1

#### 6.32.1 Detailed Description

Response to a set audio output request.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

### 6.33 FepAseSetAudioPcmFormatCommandResp Struct Reference

Response from FEP when setting PCM format.

#### Public Types

- enum Error {
  **NO_ERROR** = 1, NOT_SUPORTED_FORMAT = 2, NOT_SUPPORTED_RATE = 3, NOT_SUPPORTED_↩
  CHANNELS = 4,
  **UNKNOWN_ERROR** = 5 }

#### Public Attributes

- repeated Error error = 1

  *A list of possible error reasons.*

#### 6.33.1 Detailed Description

Response from FEP when setting PCM format.

#### 6.33.2 Member Enumeration Documentation

##### 6.33.2.1 enum FepAseSetAudioPcmFormatCommandResp::Error

**Enumerator**

***NOT_SUPORTED_FORMAT*** Signal that DSP does not support the format in the command.

***NOT_SUPPORTED_RATE*** Signal that DSP does not support the rate in the command.

***NOT_SUPPORTED_CHANNELS*** Signal that DSP does not suport the specified channels in the command.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

### 6.34 FepAseStatistics Struct Reference

Response from AseFepReq(STATISTICS)

**Classes**

- struct KeyValuePair

**Public Attributes**

- repeated KeyValuePair **keyValue** = 1

#### 6.34.1 Detailed Description

Response from AseFepReq(STATISTICS)

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

### 6.35 FepAseTunnel Struct Reference

Message intended for debugging and production test.

**Public Attributes**

- optional bytes **data** = 1

#### 6.35.1 Detailed Description

Message intended for debugging and production test.

The data included in the message can have any arbitrary format agreed between production test FEP software development to support hardware verification on the production line.

Typically the message is used to return requested information or signal an event due to UI interaction

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

### 6.36 FepAseVersionInfo Struct Reference

Response from AseFepReq(VERSION_INFO)

#### Classes

- struct Module

#### Public Attributes

- repeated Module module = 1

  *Contains version information about the connected hardware, it may consis of multiple modules.*

#### 6.36.1 Detailed Description

Response from AseFepReq(VERSION_INFO)

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

### 6.37 FepAseVolumeFadeComplete Struct Reference

Continuous volume change operation has completed.

#### Public Attributes

- required uint32 volume = 1

  *Current volume level after the fade operation.*

#### 6.37.1 Detailed Description

Continuous volume change operation has completed.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.38 InternalSpeaker Struct Reference

#### Public Types

- enum **Position** { **LEFT** = 0, **RIGHT** = 1, **CENTRE** = 2 }
- enum **Type** { **TWEETER** = 0, **MIDRANGE** = 1, **WOOFER** = 2, **FULLRANGE** = 3 }

#### Public Attributes

- required Position position = 1

  *Valid combinations for FS1: CENTRE - WOOFER CENTRE - FULLRANGE.*
- required Type **type** = 2
- optional double compensation = 3

  *The amount of gain compensation to apply to the speaker unit in dB.*

#### 6.38.1 Member Data Documentation

##### 6.38.1.1 required Position InternalSpeaker::position = 1

Valid combinations for FS1: CENTRE - WOOFER CENTRE - FULLRANGE.

Valid combinations for FS2: CENTRE - WOOFER LEFT - MIDRANGE RIGHT - MIDRANGE CENTRE - TWEETER

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.39 FepAseStatistics::KeyValuePair Struct Reference

#### Public Attributes

- optional string key = 1

  *Unique identifier of the value.*
- optional string value = 2

  *Counter, timer or other statistical information.*

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.40 FepAseVersionInfo::Module Struct Reference

### Public Attributes

- optional string **name** = 1
- optional string hardware = 2

  *The hardware version is in human readable format.*
- optional string bootloader = 3

  *Software version format shall consist of four numbers in a format: X.Y.Z.W Although using the numbers is project dependent, a general principle is following: "X" would be increased when major changes are introduced.*
- optional string **application** = 4
- optional string configuration = 5

  *If the module can operate in multiple configurations e.g.*

### 6.40.1 Member Data Documentation

#### 6.40.1.1 optional string FepAseVersionInfo::Module::bootloader = 3

Software version format shall consist of four numbers in a format: X.Y.Z.W Although using the numbers is project dependent, a general principle is following: "X" would be increased when major changes are introduced.

For example, when the software is incompatible with previous versions of hardware or software. "Y" would be changed when there are major changes in the software, adding new features for example. "Z" would be changed when there are minor changes in existing software features. "W" would be for project-specific usage (e.g. an SVN revision number).

#### 6.40.1.2 optional string FepAseVersionInfo::Module::configuration = 5

If the module can operate in multiple configurations e.g.

the DSP, information of the current information is returned.

#### 6.40.1.3 optional string FepAseVersionInfo::Module::hardware = 2

The hardware version is in human readable format.

Can contain any relevant information about the hardware such as version, BOM, PCB, variant and MAC.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.41 NetworkInfo Struct Reference

### Public Types

- enum **Type** { **WIFI** = 1, **ETHERNET** = 2, **SOFT_AP** = 3 }
- enum State {
  UNKNOWN = 0, SCANNING = 1, CONNECTING = 2, AUTHENTICATING = 3,
  ACQUIRING = 4, CONNECTED = 5, DISCONNECTED = 6, FAILED = 7 }

#### Public Attributes

- optional Type **type** = 1
- optional State **state** = 2
- optional WiFi wifi = 3

  *WiFi member is only present when type equals WIFI.*

### 6.41.1 Member Enumeration Documentation

#### 6.41.1.1 enum NetworkInfo::State

#### Enumerator

***UNKNOWN*** The device's state is unknown.

***SCANNING*** Searching for an available access point.

***CONNECTING*** Currently setting up data connection.

***AUTHENTICATING*** Network link established, performing authentication.

***ACQUIRING*** Awaiting response from DHCP server in order to assign IP address information.

***CONNECTED*** IP traffic should be available.

***DISCONNECTED*** IP traffic not available.

***FAILED*** Attempt to connect failed.

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.42 PositionSoundMode Struct Reference

#### Public Types

- enum **Position** {
  **UNDEFINED** = 0, **FREE** = 1, **WALL** = 2, **CORNER** = 3,
  **TABLE** = 4 }
- enum **Orientation** { **NONE** = 0, **HORIZONTAL** = 1, **VERTICAL** = 2 }

#### Public Attributes

- optional Position **position** = 1
- optional Orientation **orientation** = 2

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

## 6.43 WiFi Struct Reference

#### Public Types

- enum **Quality** { **EXELENT** = 0, **GOOD** = 1, **POOR** = 2 }

#### Public Attributes

- optional bool Configured = 1

  *When true, WiFi setup has been performed.*
- optional Quality quality = 2

  *Connection quality. Because RSSI is vendor specific it is converted to the levels specified by Quality.*

The documentation for this struct was generated from the following file:

- beo_ase_fep.proto

# 7 File Documentation

## 7.1 beo_ase_fep.proto File Reference

#### Classes

- struct FepAseCommand

  *Command request from FEP to ASE (Local UI).*
- struct FepAseEvent

  *Event from FEP to ASE.*
- struct FepAseReq

  *Request from FEP to ASE.*
- struct AseFepEvent

  *Event from ASE to FEP.*
- struct AseFepVolumeChangedEvent

  *Event from ASE to FEP with new volume value, the speakers output volume is set to this value.*
- struct FepAseAbsoluteVolumeCommand

  *Command from FEP to ASE to change volume to a specific level, the requested volume change does not take effect until AseFepVolumeChangedEvent is received by FEP.*
- struct FepAseRelativeVolumeCommand

  *Command from FEP to ASE to change the volume relative to the current volume level, the requested volume change does not take effect until AseFepVolumeChangedEvent is received by FEP.*
- struct AseFepVolumeFadeEvent

  *Perform a continuous volume change from one volume level to another volume level in a defined time frame.*
- struct FepAseVolumeFadeComplete

  *Continuous volume change operation has completed.*
- struct AseFepSetLineInSensitivity

  *Set the line-in sensitivity level.*
- struct AseFepSetAudioPcmFormatCommand
- struct FepAseSetAudioPcmFormatCommandResp

  *Response from FEP when setting PCM format.*
- struct FepAseAudioSocketChangeEvent

  *Event from FEP to ASE when a plug, e.g. mini-jack, have been inserted or removed from a (audio) socket.*
- struct AseFepTunnel

  *Message intended for debugging and production test.*
- struct FepAseTunnel

  *Message intended for debugging and production test.*
- struct AseFepSetAudioInputReq

*Request to set audio input.*

- struct FepAseSetAudioInputResp

  *Response to a set audio input request.*
- struct AseFepSetAudioOutputReq

  *Request to set audio output.*
- struct FepAseSetAudioOutputResp

  *Response to a set audio output request.*
- struct FepAseAliveReq

  *Request the ASE to give proof of life.*
- struct AseFepAliveResp

  *Alive message returned to FEP on request.*
- struct FepAseProductLog
- struct FepAseServiceLog

  *Write an entry in the service log.*
- struct PositionSoundMode
- struct AseFepSetPositionSoundModeReq

  *Sets the position sound mode.*
- struct AseFepGetPositionSoundModeReq

  *Gets the current selected position sound mode.*
- struct FepAsePositionSoundModeResp

  *Response when setting position sound mode.*
- struct InternalSpeaker
- struct AseFepSetInternalSpeakerCompensationCommand

  *Set the internal speaker compensation gain for one or more speaker units.*
- struct FepAseInternalSpeakerCompensationResp

  *Response when setting or getting internal speaker compensation.*
- struct FepAsePlayComfortToneCmd

  *Instruct ASE to play a comfort tone.*
- struct AseFepReq

  *Send request from ASE to FEP.*
- struct FepAseProductInfo

  *Response from AseFepReq(PRODUCT_INFO)*
- struct FepAsePowerStatus

  *Sent upon AseFepReq(POWER_STATUS) request and when power/battery status changes.*
- struct FepAseVersionInfo

  *Response from AseFepReq(VERSION_INFO)*
- struct FepAseVersionInfo::Module
- struct FepAseStatistics

  *Response from AseFepReq(STATISTICS)*
- struct FepAseStatistics::KeyValuePair
- struct WiFi
- struct NetworkInfo
- struct AseFepNetworkInfo

  *Current network status information. Sent when network status has changed or upon request from FEP.*
- struct AseFepMessage
- struct FepAseMessage

# Index