



**Research & Vehicle Technology**  
**“Infotainment Systems Product Development”**

**Feature – Alerts v4**  
**(Chimes/Prompts/Beeps/Audio Attenuation)**

**Infotainment Subsystem Part Specific**  
**Specification (SPSS)**

Version 1.4

**UNCONTROLLED COPY IF PRINTED**

Version Date: February 20, 2023

**FORD CONFIDENTIAL**



## Revision History

Date	Version	Notes	
August 5, 2021	1.0	Initial Release	
November 9, 2021	1.1		
	ALERTv2-SR-REQ-413887/B-Infotainment Chime Attenuation - Phoenix Architecture		jmyslin2: added clarification
	ALERT-SR-REQ-052682/L-Chime Requests while the infotainment system is muted during a cold crank		jmyslin2: Updated requirement for BEV vehicles to not mute chimes for crank CAN signals
	ALERTv3-SR-REQ-413987/B-Chime Error States / Fault handling - variant 3		jmyslin2: added update on PDC_Chime_Supported being set to inactive
February 10, 2022	1.2		
	ALERT-SR-REQ-478177/A-Chime ID Assignments - variant 2 (Phoenix)		jmyslin2: new Chime ID Assignments requirement for Phoenix architecture
September 27, 2022	1.3		
	ALERTv2-SR-REQ-521337/A-Power-up time for infotainment components (Phoenix architecture)		jmyslin2: new requirement replacing REQ-014735
February 20, 2023	1.4		
	ALERT-SR-REQ-052682/M-Chime Requests while the infotainment system is muted during a cold crank		jmyslin2: updated the requirement per Dave Walus to change the crank unmute time to 500 msec



# Table of Contents

REVISION HISTORY .....	2
<b>1 ARCHITECTURAL DESIGN.....</b>	<b>4</b>
1.1 Definitions / Acronyms.....	4
1.2 ALERT-CLD-REQ-014717/C-Chime Client (TcSE ROIN-202535-1).....	4
1.3 ALERT-CLD-REQ-014718/B-Chime Generator Server (TcSE ROIN-202536-1) .....	4
1.4 ALERT-CLD-REQ-014719/B-Chime Audio Source Server (TcSE ROIN-202537-1).....	4
1.5 Interface Requirements .....	4
1.5.1 MD-REQ-275630/B-IPC_Infotainment .....	4
1.5.2 MD-REQ-275638/C-IPC_Chime .....	4
1.5.3 MD-REQ-275692/C-IPC_Chime2 .....	6
1.5.4 MD-REQ-413858/A-PDC_Alert .....	8
1.5.5 MD-REQ-413860/A-AHU_Alert - variant 2.....	9
1.5.6 MD-REQ-413861/A-DSPAMP_Alert - variant 2 .....	9
1.5.7 MD-REQ-413859/A-SYNC_Alerts - variant 2.....	10
1.5.8 ALERT-MD-REQ-232947/E-rearParkAidChimeVolumeSelection - FBMP .....	10
1.5.9 ALERT-MD-REQ-233050/C-frontParkAidChimeVolumeSelection - FBMP .....	11
<b>2 FUNCTIONAL DEFINITION .....</b>	<b>12</b>
2.1 ALERTv3-FUN-REQ-413857/A-Chimes - variant 3 (integrated Chime Client, Chime Generator and Chime Source) ..	12
2.1.1 Overview.....	12
2.1.2 Deployment chimes variant 3 .....	12
2.1.3 Chime Activation Requirements .....	13
2.1.4 Chime Error Management Requirements .....	23
2.1.5 Sequence Diagrams .....	26
2.2 Adjustable Chimes Deployment .....	37
2.3 ALERT-FUN-REQ-195348/D-Adjustable Chime Strategy .....	37
2.3.1 Use Cases .....	37
2.3.2 Requirements .....	42
2.3.3 Sequence Diagrams .....	46
2.4 ALERTv3-FUN-REQ-414038/A-Mixable Prompts - Variant 3.....	49
2.4.1 Overview - mixable prompts.....	49
2.5 ALERTv2-FUN-REQ-414037/A-Beeps .....	50
2.5.1 Overview - Beeps .....	50
2.6 ALERT-FUN-REQ-414530/A-Audio Attenuation Phoenix Architecture - Captains Announcement and ICC .....	51
2.6.1 Requirements .....	51
2.7 ALERT-FUN-REQ-014794/D-Audio Attenuation/Muting Strategy (TcSE ROIN-119789-1) .....	52
2.7.1 Cluster Prioritizing Audio Attenuation Requests .....	52
2.7.2 AHU / DSP AMP Prioritizing Audio Attenuation Requests.....	52
2.7.3 Infotainment System Audio Attenuation .....	52
<b>3 APPENDIX: REFERENCE DOCUMENTS.....</b>	<b>56</b>



# 1 Architectural Design

## 1.1 Definitions / Acronyms

PDC - Phoenix Domain Controller

PAC - Phoenix Audio Controller

AHU - Audio Head Unit

APIM - Auxiliary Protocol Interface Module (ie SYNC)

DSP AMP – Digital Signal Processor external smart amplifier module

IPC – Instrument Panel Cluster

## 1.2 ALERT-CLD-REQ-014717/C-Chime Client (TcSE ROIN-202535-1)

The Chime Client is the chime master that tells the infotainment system what chime(s) to play

## 1.3 ALERT-CLD-REQ-014718/B-Chime Generator Server (TcSE ROIN-202536-1)

The Chime Generator Server produces the chime signal for a respective chime

## 1.4 ALERT-CLD-REQ-014719/B-Chime Audio Source Server (TcSE ROIN-202537-1)

The Chime Audio Source Server produces the chime sound heard in the infotainment system

## 1.5 Interface Requirements

### 1.5.1 MD-REQ-275630/B-IPC\_Infotainment

**Message Type:** Status

Method sent from the Chime Client to the Infotainment modules for Chime Power Moding, Chime Audio Source and Chime Audio Attenuation

Logical Signal Name	Literals	Value	Description
Chime_Source	Inactive/Invalid	0x0	Chime audio source signal
	Infotainment_System	0x1	
	Cluster	0x2	
Power_Up_Chime_Modules	Inactive	0x0	Chime power moding signal
	Active	0x1	
Attn_Info_Audio	No Attenuation of Audio	0x0	Audio Attenuation signal
	Attenuation_1	0x1	
	Attenuation_2	0x2	
	cont.		
	Attenuation_6	0x6	
	Unknown	0x7	
New_Attn_Event	Inactive	0x0	New attenuation event signal
	Active	0x1	

### 1.5.2 MD-REQ-275638/C-IPC\_Chime

**Message Type:** Request



Request from the Chime Client to the Infotainment chime modules (Chime Generator Server and Chime Audio Source Server) to play a particular chime

Logical Signal Name	Literals	Value	Description
Chime	Inactive	0x0	Chime Sound ID
	No_Chime	0x1	
	Chime_1	0x2	
	Chime_2	0x3	
	Chime_3	0x4	
	cont.		
	<a href="#">Chime_254</a>	<a href="#">0xFF</a>	
Chime_Vol_Level	100% of stored chime volume / Inactive	0x0	Chime Volume Level
	99% of stored chime volume	0x1	
	98% of stored chime volume	0x2	
	97% of stored chime volume	0x3	
	cont.		
	2% of stored chime volume	0x62	
	1% of stored chime volume	0x63	
	Reserved	0x64	
Chime_Directionality	OFF / Inactive	0x0	Chime Speaker Directionality
	All	0x1	
	Front	0x2	
	Rear	0x3	
Chime_Time_Criticality	Inactive	0x0	Chime Criticality
	Criticality_High	0x1	
	Criticality_Low	0x2	
OFF_Time_Btwn_Chime	Inactive	0x0	Off Time Between Chimes
	Continuously (repeat continuously)	0x1	
	5 msec	0x2	
	10 msec	0x3	
	15 msec	0x4	
	20 msec	0x5	
	40 msec	0x6	
	60 msec	0x7	
	conti... increment in 20 msec until 800 msec		
	800 msec	0x2C	
	850 msec	0x2D	
	900 msec	0x2E	
	1 sec	0x2F	
	1.5 sec	0x30	
	2.0 sec	0x31	
	3 sec	0x32	
	4 sec	0x33	
	5 sec	0x34	
Chime_Occurence	Inactive	0x0	Determines the number of repetitions the Chime_X sound to be played, or for the Chime_X to be played continuously
	Play Once	0x1	
	2 Repetitions	0x2	



	3 Repetitions	0x3	
	4 Repetitions	0x4	
	5 repetitions	0x5	
	cont...		
	14 repetitions	0xE	
	Continuous	0xF	
AdjustableChimeVol	Default Volume / Inactive / 0 adjustable chime volume level	0x0	Adjustable Chime Volume Level
	-6 Reserved, do not use	0x1	
	-5 Reserved, do not use	0x2	
	-4 Reserved, do not use	0x3	
	-3 adjustable chime volume level	0x4	
	-2 adjustable chime volume level	0x5	
	-1 adjustable chime volume level	0x6	
	+1 adjustable chime volume level	0x7	
	+2 adjustable chime volume level	0x8	
	+3 adjustable chime volume level	0x9	
	+4 Reserved, do not use	0xA	
	+5 Reserved, do not use	0xB	
	+6 Reserved, do not use	0xC	
	Reserved, do not use	0xD	
	Reserved, do not use	0xE	
	Reserved, do not use	0xF	
ChimeID_No_Rq	Inactive	0x0	Chime ID
	ID 1	0x1	
	ID 2	0x2	
	ID 3	0x3	
	cont.		
	ID 255	0xFF	

**1.5.3 MD-REQ-275692/C-IPC\_Chime2****Message Type:** Request

Request from the Chime Client to the Infotainment chime modules (Chime Generator Server and Chime Audio Source Server) to play a particular chime

Logical Signal Name	Literals	Value	Description
Chime2	Inactive	0x0	Chime Sound ID
	No_Chime	0x1	
	Chime_1	0x2	
	Chime_2	0x3	
	Chime_3	0x4	
	cont.		



	<u>Chime_254</u>	<u>0xFF</u>	
Chime_Vol_Level2	100% of stored chime volume / Inactive	0x0	Chime Volume Level
	99% of stored chime volume	0x1	
	98% of stored chime volume	0x2	
	97% of stored chime volume	0x3	
	cont.		
	2% of stored chime volume	0x62	
	1% of stored chime volume	0x63	
	Reserved	0x64	
Chime_Directionality2	OFF / Inactive	0x0	Chime Speaker Directionality
	All	0x1	
	Front	0x2	
	Rear	0x3	
Chime_Time_Criticality2	Inactive	0x0	Chime Criticality
	Criticality_High	0x1	
	Criticality_Low	0x2	
OFF_Time_Btw_Chime 2	Inactive	0x0	Off Time Between Chimes
	Continuously (repeat continuously)	0x1	
	5 msec	0x2	
	10 msec	0x3	
	15 msec	0x4	
	20 msec	0x5	
	40 msec	0x6	
	60 msec	0x7	
	conti... increment in 20 msec until 800 msec		
	800 msec	0x2C	
	850 msec	0x2D	
	900 msec	0x2E	
	1 sec	0x2F	
	1.5 sec	0x30	
	2.0 sec	0x31	
	3 sec	0x32	
	4 sec	0x33	
	5 sec	0x34	
Chime_Occurence2	Inactive	0x0	Determines the number of repetitions the Chime2_X sound to be played, or for the Chime2_X to be played continuously
	Play Once	0x1	
	2 Repetitions	0x2	
	3 Repetitions	0x3	
	4 Repetitions	0x4	
	5 repetitions	0x5	
	cont...		
	14 repetitions	0xE	
AdjustableChimeVol2	Continuous	0xF	Adjustable Chime Volume Level
	Default Volume / Inactive / 0 adjustable chime volume level	0x0	
	-6 Reserved, do not use	0x1	



	-5 Reserved, do not use	0x2	
	-4 Reserved, do not use	0x3	
	-3 adjustable chime volume level	0x4	
	-2 adjustable chime volume level	0x5	
	-1 adjustable chime volume level	0x6	
	+1 adjustable chime volume level	0x7	
	+2 adjustable chime volume level	0x8	
	+3 adjustable chime volume level	0x9	
	+4 Reserved, do not use	0xA	
	+5 Reserved, do not use	0xB	
	+6 Reserved, do not use	0xC	
	Reserved, do not use	0xD	
	Reserved, do not use	0xE	
	Reserved, do not use	0xF	
Chime2ID_No_Rq	Inactive	0x0	Chime ID
	ID 1	0x1	
	ID 2	0x2	
	ID 3	0x3	
	cont.		
	ID 255	0xFF	

**1.5.4 MD-REQ-413858/A-PDC\_Alert****Message Type:** Status

APIM PDC (Phoenix Domain Controller) Alert Status from the Chime Generator & Chime Audio Source to the Chime Client / Chime Master

Logical Signal Name	Literals	Value	Description
PDC_Chimes_Supported	Inactive	0x0	Signal from the indicating to the Chime Master (ie Cluster) if chimes are supported by the Infotainment system
	Supported	0x1	
	Not_Supported	0x2	
PDC_Chime_Not_Recognized	Inactive	0x0	Signal from the Chime Generator to Cluster letting the Cluster know the chime requested is not recognized. If the Chime is recognized the signal remains set to inactive.
	Chime_Not_Recognized	0x1	
PDC_Chime2_Not_Recognized	Inactive	0x0	Signal from the Chime Generator to the Cluster Chime Client indicating what chime is active.
	Chime_Not_Recognized	0x1	
PDC_Chime_Active	Inactive	0x0	
	No_Chime	0x1	
	Chime_1	0x2	
	Chime_2	0x3	
	Chime_3	0x4	
	cont.		
	<a href="#">Chime_254</a>	<a href="#">0xFF</a>	
PDC_Chime2_Active	Inactive	0x0	





	No_Chime	0x1	
	Chime_1	0x2	
	Chime_2	0x3	
	Chime_3	0x4	
	cont.		
	<a href="#">Chime_254</a>	<a href="#">0xFF</a>	
PDC_AdjustableChimeVolSupported	Inactive	0x0	Signal to indicate if a requested adjustable Chime value (ex +1, -2) for a particular Chime_X sound is supported or not
	Adjustable Chime Sound Not Supported	0x1	
	Adjustable Chime Sound Supported	0x2	
PDC_AdjustableChimeVolSupported 2	Inactive	0x0	
	Adjustable Chime Sound Not Supported	0x1	
	Adjustable Chime Sound Supported	0x2	
PDC_ChimeID_No_Stat	Inactive	0x0	Signal for Chime ID
	ID1	0x1	
	ID2	0x2	
	ID3	0x3	
	cont.		
	ID255	0xFF	
PDC_Chime2ID_No_Stat	Inactive	0x0	
	ID1	0x1	
	ID2	0x2	
	ID3	0x3	
	cont.		
	ID255	0xFF	

**1.5.5 MD-REQ-413860/A-AHU\_Alert - variant 2****Message Type:** Status

Alert Status from the Audio Head Unit with the Chime Supported status

Logical Signal Name	Literals	Value	Description
AHU_Chimes_Supported	Inactive	0x0	Signal from the AHU indicating to the if chimes are supported by the AHU
	Supported	0x1	
	Not_Supported	0x2	

**1.5.6 MD-REQ-413861/A-DSPAMP\_Alert - variant 2****Message Type:** Status

Alert Status from the DSP AMP with the Chime Supported status

Logical Signal Name	Literals	Value	Description
DSP_Chimes_Supported	Inactive	0x0	Signal indicating if chimes are supported by the DSP AMP
	Supported	0x1	
	Not_Supported	0x2	

**1.5.7 MD-REQ-413859/A-SYNC\_Alerts - variant 2****Message Type:** Status

Signal from APIM Phoenix Domain Controller for Audio Attenuation

Logical Signal Name	Literals	Value	Description
Attn_Info_Audio	No Attenuation of Audio	0x0	
	Attenuation_1	0x1	
	Attenuation_2	0x2	
	Attenuation_3	0x3	
	cont.		
	Attenuation_6	0x6	
	Unknown	0x7	
New_Attn_Event	Inactive	0x0	
	Active	0x1	
	Active	0x1	

**1.5.8 ALERT-MD-REQ-232947/E-rearParkAidChimeVolumeSelection - FBMP**

For this feature when performing the “Set” or “Query” operation the Feature Number and Configuration Number in the Feature.Rq and Feature.St messages shall be used below.

If Enhanced Memory is supported the Active Personality Profile shall be used for PersIndex. If Enhanced Memory is not supported PersIndex shall be set to Vehicle. See Feature Based Message Protocol SPSS for details.

Feature Description	Feature Number	Configuration Number	Configuration Name / HMI selection
Rear Park Aid Adjustable Volume Selection and Status  rearParkAidChimeVolSelection signal in SPSS	0x0910	0x00	-6 Reserved / Do not use
		0x01	-5 Reserved / Do not use
		0x2	-4 Reserved / Do not use
		0x3	-3 adjustable volume level
		0x4	-2 adjustable volume level
		0x5	-1 adjustable volume level
		0x6	0 Adjustable chime volume / Default Volume
		0x7	+1 adjustable volume level
		0x8	+2 adjustable volume level
		0x9	+3 adjustable volume level
		0xA	+ 4 Reserved / Do not use
		0xB	+ 5 Reserved / Do not use
		0xC	+ 6 Reserved / Do not use

**1.5.9 ALERT-MD-REQ-233050/C-frontParkAidChimeVolumeSelection - FBMP**

For this feature when performing the “Set” or “Query” operation the Feature Number and Configuration Number in the Feature.Rq and Feature.St messages shall be used below.

If Enhanced Memory is supported the Active Personality Profile shall be used for PersIndex. If Enhanced Memory is not supported PersIndex shall be set to Vehicle. See Feature Based Message Protocol SPSS for details.

Feature Description	Feature Number	Configuration Number	Configuration Name / HMI selection
Front Park Aid Adjustable Volume Selection and Status  frontParkAidChimeVolSelection signal in SPSS	0x0911	0x00	-6 Reserved / Do not use
		0x01	-5 Reserved / Do not use
		0x2	-4 Reserved / Do not use
		0x3	-3 adjustable volume level
		0x4	-2 adjustable volume level
		0x5	-1 adjustable volume level
		0x6	0 Adjustable chime volume / Default Volume
		0x7	+1 adjustable volume level
		0x8	+2 adjustable volume level
		0x9	+3 adjustable volume level
		0xA	+ 4 Reserved / Do not use
		0xB	+ 5 Reserved / Do not use
		0xC	+ 6 Reserved / Do not use



## 2 Functional Definition

### 2.1 ALERTv3-FUN-REQ-413857/A-Chimes - variant 3 (integrated Chime Client, Chime Generator and Chime Source)

#### 2.1.1 Overview

For Chimes variant 3 the feature integrates the Chime Client/Cluster, Chime Generator and Chime Audio Source functionality into one Phoenix Domain Controller module and they are no longer separate modules on the CAN bus. The Cluster and Chime Generator / Chime Audio Source functionality are separate core processors at the time this spec was written so an internal interface must still be developed. From this spec the logical signals could be used to develop signals internal to APIM PDC (ex logical IPC\_Chime message between Chime Client (ie VMCU) and Chime Generator/Chime Audio Source (ie CCPU).

The AHU and DSP AMP are no longer part of chime command and control (ie they don't look at IPC\_Chime/IPC\_Chime2 messages). Chime audio is simply passed from the APIM PDC (Phoenix Domain Controller) module to the AHU / DSP AMP. The AHU / DSP AMP has audio routed to the speakers connected to it so chime diagnostics must still be performed.

Note:

If the sequences diagrams reference CAN that should be ignored as the sequence diagrams should be considered at the logical level (ie not network dependent)

#### 2.1.2 Deployment chimes variant 3

The table below shows how the logical classes may be mapped to physical modules for the Chime feature. The table below covers the lead program.

At the time the specification was written the below table was the latest. If there are additional modules deployed to the class descriptions or the vehicle architecture changed since the spec was written and released, then the applicable implementation guide class description would cover those modules. If there is a conflict between the implementation guide and the table below the implementation guide takes precedent.

Logical Class	APIM PDC	APIM PDC VMCU	APIM PDC CCPU	DSP AMP variant 2	AHU variant 2 / PAC
Chime Client (Chime Master)	x	x			
Chime Generator *See note 1	x		x		
Chime Audio Source	x		x		
Back-up / early Chime Generator	x	x			
Chime Arbitrator	x	x			
Chime Diagnostics	x		x	x	x

- X – represents functionality supported by that module
- Blank – represents functionality not supported by that module

Note 1:

You might see Chime Generator or AHU Chime Generator in the spec. The table above shall be followed even though it might say AHU next to Chime Generator it is not the AHU for Phoenix but the APIM PDC. The AHU is just from legacy requirements.



## Note 2:

For the Phoenix AHU/PAC and DSP AMP the Chime requirements will only apply if it says AHU variant 2 / PAC or DSP AMP variant 2 in the requirement or if selected yes in the implementation guide

## Note 3:

In this spec if legacy requirements reference Cluster then this references the APIM PDC module or chime client / chime master (ie VMCU)

### 2.1.3 Chime Activation Requirements

#### 2.1.3.1 ALERTv2-SR-REQ-413887/B-Infotainment Chime Attenuation - Phoenix Architecture

For Chimes variant 3, the AHU variant 2 and DSP AMP variant 2 do not act upon the IPC Infotainment chime audio attenuation signals (ie Attn\_Info\_Audio & New\_Attn\_Event) unless specifically noted otherwise.

The APIM CCPU receives the chime audio attenuation internally from the Cluster VMCU and the APIM CCPU sends the volume status signals (ex Audio\_Vol\_Level) to the AHUv2 and DSP AMPv2 at the attenuated level to reflect the chime audio attenuation.

In zone mode (if supported – VehicleAudioMode = Zone) the chime audio attenuation/mute shall apply to all zones if playing audio out of a vehicle speaker.

#### 2.1.3.2 ALERT-SR-REQ-014733/C-Infotainment Components Power Mode signals (TcSE ROIN-40352-1)

The Infotainment Chime Components shall be capable of producing chimes from Chime Requests (IPC\_Chime / IPC\_Chime2) when:

'Power\_Up\_Chime\_Modules == Active' AND the Infotainment System is the Chime Source 'IPC\_Infotainment.St() : Chime\_Source == Infotainment\_System'.

The infotainment components responsible for chimes shall be capable of producing chimes regardless of the HMIAudioMode status when Chimes is enabled through the Infotainment System.

#### 2.1.3.3 ALERTv2-SR-REQ-415053/A-Chime Power Mode signal usage

The Cluster shall wake and keep up the Infotainment bus (if not already awake) and set 'Power\_Up\_Chime\_Modules = Active' for predictive triggers (ex. open door) as defined in the Cluster engineering specification.

An exception would be for error states such as the APIM PDC setting PDC\_Chime\_Supported = Not\_Supported where the Cluster would set Power\_Up\_Chime\_Modules = Inactive when it receives Not\_Supported.

When the predictive triggers are no longer valid then the signal 'Power\_Up\_Chime\_Modules = Inactive'.

#### 2.1.3.4 ALERTv2-SR-REQ-521337/A-Power-up time for infotainment components (Phoenix architecture)

The infotainment chime components shall be capable of producing chime audio within 2.0 seconds of 'Power\_Up\_Chime\_Modules = Active'.

- Phoenix only: the APIM AOS Chime Generator shall be capable of producing chime audio within 5 seconds of 'Power\_Up\_Chime\_Modules = Active' coming out of a cold boot (ie not coming out of suspend to RAM). Normal operation from suspend to RAM shall still happen within 2.0 seconds.

For supporting time to first infotainment chime audio reference the Infotainment Diagnostic Specification (IDS) for chime diagnostics start-up using the Power\_Up\_Chime\_Module signal.

#### 2.1.3.5 ALERT-SR-REQ-014737/C-Timing for single Chime Source module to produce Chime audio (TcSE ROIN-40356-3)

When the Chime Generator and Chime Audio Source are the same Chime Source module then the Chime Source module shall start producing chime audio within 70 msec of receiving the new chime request message.



### 2.1.3.6 ALERT-SR-REQ-014741/K-Cluster Chime Requests (TcSE ROIN-40360-1)

Once the Cluster receives a chime request (from the vehicle network) it shall process and transmit to the Infotainment System within 60 msec the applicable IPC\_Chime message to the infotainment system with the chime signal parameters set.

If there is a currently playing chime the Cluster shall determine the priority of the currently playing chime to see if it should be overwritten by the new chime request. If the new chime request has a higher priority than a currently playing fixed repetitive chime then the Cluster may want to monitor the AHU\_Chime\_Active signal to see if the new chime can be delayed until the current chime is complete or if it should immediately be overwritten.

The Cluster shall set all the signals in the IPC\_Chime (1 or 2) messages and keep all the signals populated and unchanged as long as the chime is active (exception OFF\_Time\_Btwn\_Chime signal). Only when the Chime is done should the IPC\_Chime signals be set to No\_Chime and Inactive.

The Cluster shall send a No\_Chime for the currently playing chime before sending any new chime requests with the directionality for the same speakers. See sequence diagrams for examples of when chime requests are sent by the Cluster.

Note: If anything in the Cluster chime arbitrator spec contradicts what is in the chime section of the SPSS it should be brought to Ford's attention.

#### Multiple Chime Requests for speakers with the same directionality:

If the AHU is playing a chime (ex AHU\_Chime\_Active = Chime\_Y) and if the Cluster needs to end the chime and play a new chime using the same speakers/directionality, then once the Cluster sends the No\_Chime chime request to end Chime\_Y the Cluster shall wait for the AHU to respond back with "AHU\_Chime\_Active = No\_Chime" before making the new chime request. The AHU shall respond back to the Cluster No\_Chime request within 75 msec with a No\_Chime.

- Note: because of legacy Clusters the AHU cannot assume the Cluster will always wait for the AHU No\_Chime response as stated above if two chime requests occur on different alert channels (ie IPC\_Chime/IPC\_Chime2). For the AHU's if they receive a cluster chime request (Chime\_Y) on one Alert channel and then receive the No\_Chime request on the same Alert channel the Cluster could send a chime request for the same speakers/directionality on a different alert channel before the AHU responds back with a No\_Chime and the AHU still has to support. See sequence diagrams with examples.
  - [This note does not apply to Phoenix Architecture APIM PDC VCMU Cluster chime software, since this would no longer be considered a legacy Cluster.](#)
  - ~~This note does not apply to APIM 4.2 since this would no longer be considered a legacy Cluster.~~

If the AHU is not playing a chime (AHU\_Chime\_Active = No\_Chime) and if the Cluster sends a chime request such as IPC\_Chime(Chime = Chime\_X) and then the Cluster quickly sends a No\_Chime before the AHU responds to the original Chime\_X request with AHU\_Chime\_Active = Chime\_X then the Cluster shall not send a new chime request until:

- The AHU responded back with AHU\_Chime\_Active = Chime\_X and then AHU\_Chime\_Active = No\_Chime, OR
- If the AHU does not respond back to the original Chime\_X response (stays at AHU\_Chime\_Active = No\_Chime) the Cluster shall wait 75 msec before making any new chime requests

### 2.1.3.7 ALERT-SR-REQ-014742/C-Chime\_Vol\_Level signal (TcSE ROIN-40361-3)

The Chime Audio Source stores the volume levels for each of the individual chimes. The Chime Audio Source will use the Chime\_Vol\_Level signal from the Cluster to determine how much lower the chime should be played from the stored volume level.

If the stored chime volume level in the chime audio source is to be played at 100% of the stored volume level then the Cluster shall set the signal Chime\_Vol\_Level = Inactive.

Once a chime is selected by the Cluster and broadcast on the infotainment bus the Cluster shall not change the selected chime volume level until the chime has ended or until there is a new chime activated.



### 2.1.3.8 ALERT-SR-REQ-014743/C-Chime Mixing (TcSE ROIN-40362-1)

The Chime Audio Source shall be capable of mixing the Chimes together with the main audio source as defined in the applicable Chime Audio Source component engineering specifications.

### 2.1.3.9 ALERTv3-SR-REQ-413947/A-Chime Not Recognized - variant 3

#### Chime Generator:

If the Chime Generator receives a command from the Cluster to play a particular chime that the Chime Generator does not recognize (Chime\_X where X is unknown) then the Chime Generator shall send the 'XXX Chime Not Recognized = Chime Not Recognized' signal within 75msec of receiving the chime command.

Whenever the Chime Generator sets the signal Chime Not Recognized = Chime Not Recognized then at the same time in the same message the Chime\_Active signal will be used to say what Chime the Chime Generator cannot play.

#### Cluster:

The Cluster shall monitor the Chime Generators Chime Not Recognized signal and when it equals Chime Not Recognized the Cluster shall then play the requested Chime indicated in the AHU\_Chime\_Active signal and update the IPC\_Chime / IPC\_Chime2 message to reflect the chime is no longer requested for the infotainment system.

### 2.1.3.10 ALERT-SR-REQ-014746/D-OFF Time Between Chime signal (TcSE ROIN-40366-1)

The Cluster shall tell the Chime Generator how long there is no chime audio between playing a particular chime sound using the OFF\_Time\_Btwn\_Chime signal (ex. pauses between repeating Chime\_17 Reverse Park Aid sound). For the same chime this value may change.

Within a particular Chime Request from the Cluster the OFF\_Time\_Btwn\_Chime signal could change for the same chime but this should not change the state of the Chime Audio Source Alert input channel settings (ex volume, directionality). While the OFF\_Time\_Btwn\_Chime signal is changing for the same chime the Alert\_Channel would remain Initialized\_For\_Chimes.

For example for a reverse park aid chime as the vehicle backs up the OFF\_Time\_Btwn\_Chime signal could change to increase the beep rate by making the OFF\_Time\_Btwn\_Chime time a smaller value.

#### Ex:

1. RPA Chime event occurs
2. The Cluster sends IPC\_Chime/IPC\_Chime2 (Chime = RPA\_Chime; Chime\_Occurance = Continuous; OFF\_Time\_Btwn\_Chime = Zone1 Rate...)
3. The AHU starts playing the RPA chimes and sends AHU\_Alert (AHU\_Chime\_Active = RPA\_Chime).
4. The vehicle goes into another zone and sends IPC\_Chime/IPC\_Chime2 (Chime = RPA\_Chime; Chime\_Occurance = Continuous; OFF\_Time\_Btwn\_Chime = Zone2 Rate...) without sending a No\_Chime and only changes the OFF\_Time\_Btwn\_Chime signal.
5. The AHU plays the RPA chimes at the rate for Zone 2 and continues to send (AHU\_Chime\_Active = RPA\_Chime).

Note: in the above example both the Cluster and AHU did not send a No\_Chime when only the OFF\_Time\_Btwn\_Chime signal changed.

### 2.1.3.11 ALERT-SR-REQ-014747/J-Chime Time Criticality = Criticality\_High (TcSE ROIN-40367-1)

The Cluster shall set 'Chime\_Time\_Criticality = Criticality\_High' if a new Chime is a chime that needs the currently playing chime sound (ex Chime\_8 Ford DNA B) to be immediately ended so the new Chime can be played as quickly as possible (ex. FCW).

If the Chime Generator received a Criticality\_High Cluster No\_Chime Chime Request to end the currently playing chime then if the chime sound is playing (ex DNA B) the Chime Generator shall immediately end playing that chime sound. See AHU hardware spec for what is the longest it can take to end a chime sound with no distortions or pops ([ex 10 msec or whatever defined in AHU hardware spec](#)).





If the Chime Generator received a criticality low No\_Chime request from the Cluster to end the current chime and while the Chime Generator is still playing the current chime sound (ex AHU\_Chime\_Active = Chime\_X) if the Chime Generator then receives a criticality\_high chime request for the same speakers from the Cluster then the Chime Generator shall immediately end the current chime (ie immediately end Chime\_X sound). See sequence diagrams for detailed examples.

Ex.

Pre-Condition:

Infotainment Chime Generator and Chime Audio Source are playing a continuous chime.

Event:

Chime Clients ends the chime with Criticality\_High No Chime event. Chime Client sends IPC\_Chime (Chime = No Chime, Chime\_Time\_Criticality = Criticality\_High, Chime\_Vol\_Level = Inactive; Chime\_Directionality = Inactive; OFF\_Time\_Btwn\_Chime = Inactive; Chime\_Occurrence = Inactive)

Post-Condition:

The Chime Generator ends it currently playing Chime\_X sound (ex DNB) before it is finished and immediately sets AHU\_Chime\_Active = No Chime

2.1.3.12 ALERT-SR-REQ-014748/I-Chime\_Time\_Criticality = Criticality\_Low (TcSE ROIN-40368-1)

The Cluster shall set 'Chime\_Time\_Criticality = Criticality Low' if a new Chime is a chime that can wait for the currently playing chime to finish playing the chime sound (ex. finish Ford DNA B sound) before playing the new chime.

If the Chime Generator receives a Criticality\_Low Cluster No\_Chime request to end the current chime then if the Chime Generator is playing a chime sound (ex Ford DNA B sound) it shall finish playing the chime sound and then end the chime (ie set AHU\_Chime\_Active = No\_Chime).

If the Chime Generator received a criticality\_low Cluster chime request "Chime = No\_Chime" to end the current chime (ex Chime\_X) and while the Chime Generator is still playing the previous chime sound (ex AHU\_Chime\_Active = Chime\_X) the Chime Generator received a new low criticality Chime request for the same speakers then the AHU / DSP AMP shall start to play the new chime after playing the currently playing chime sound finishes (the AHU/DSP AMP shall play the new chime as long as the Cluster has not sent the No chime for the new chime request (ex. IPC\_Chime/IPC\_Chime2 (Chime = Chime\_New → Chime = No\_Chime)) by the time the AHU/DSP AMP finishes playing the old chime sound (Chime\_X)). See sequence diagrams for examples.

Example 1: If the Chime\_Occurrence was set to 8 repetitions of the Chime\_X sound (ex Ford DNA B) and the 3rd repetition Chime\_X sound is currently being played when the new Chime\_Y is requested with Chime\_Time\_Criticality = Criticality Low then the Chime Generator would finish the 3<sup>rd</sup> repetition of the Chime\_X sound and keep AHU\_Chime\_Active = Chime\_X until finished but not play the remaining 5 repetitions but instead switch to the new chime. See sequence diagrams for detailed examples.

Note: The Cluster can always just wait for the chime to complete it's repetitions by monitoring the AHU\_Chime\_Active = Chime\_X signal to avoid having to interrupt a chime.

Example 2.

Pre-Condition:

Infotainment Chime Generator and Chime Audio Source are playing a continuous chime (AHU\_Chime\_Active = Chime\_X).

Event:

Chime Clients ends the chime with Criticality\_Low No Chime event. Chime Client sends IPC\_Chime (Chime = No Chime, Chime\_Time\_Criticality = Criticality\_Low, Chime\_Vol\_Level = Inactive; Chime\_Directionality = Inactive; OFF\_Time\_Btwn\_Chime = Inactive; Chime\_Occurrence = Inactive)

Post-Condition:

The Chime Generator ends it currently playing Chime\_X sound (ex DNA B) after the Chime Generator is finished producing the Chime\_X sound and sets ChimeGenerator\_Chime\_Active = No\_Chime after it is done producing the sound





### 2.1.3.13 ALERTv2-SR-REQ-348582/C-Chime Directionality signal - variant 2

This variant 2 of the Chime Directionality requirement applies if the Chime Generator supports Directionality All chime along with another chime concurrently:

- The Chime Generator, if it uses one polyphonic and one simple chime generator, shall follow variant 1 of this requirement (Alert-REQ-014749-Chime Directionality signal) which does not allow directionality All chime along with another chime.
- The Chime Generator, if it uses stereo digital media chimes for all chimes, then this requirement shall be followed (allow a chime with directionality All and another Front or Rear chime to be played at the same time).

Chime Client making Directionality All chime request along with another chime request:

The Chime Client (ex Cluster) shall not request a directionality All chime request with another chime request (ie Front or Rear) at the same time until paired with a Chime Generator and Chime Audio Source that supports this requirement (directionality All plus another Front or Rear chime). If the Cluster does request this on a chime generator that does not support this requirement, then the AHU would set Chime\_Not\_Recognized for the second chime and it would be played by the Cluster.

The Cluster shall tell the Chime Generator and Chime Audio Source what speakers to play the chime through using the Chime\_Directionality signal.

- Note throughout this requirement what applies to Chime\_Directionality signal also applies to the Chime\_Directionality2 signal

Chime\_Directionality signal:

- If Chime\_Directionality = Front then the requested chime plays out of the front speakers
- If Chime\_Directionality = Rear then the requested chime plays out of the rear speakers
- If Chime\_Directionality = All then the requested chime plays out of all speakers

Allowable Combinations that the Cluster can send:

1. 1 chime Front speakers only (ex Chime\_Directionality = Front)
2. 1 chime Rear speakers only (ex Chime\_Directionality = Rear)
3. 1 chime out for Front speakers and one chime out of Rear speakers at the same time
4. 1 chime out of All speakers (ex Chime\_Directionality = All) and no other chimes
5. 1 chime out of All speakers and one chime out of the Front speakers at the same time
6. 1 chime out of All speakers and one chime out of the Rear speakers at the same time (ex Chime\_Directionality = All and Chime\_Directionality2 = Rear)

Vehicle with Front Speakers Only (no rear speakers):

For this vehicle set-up Chime\_Directionality = Rear and Chime\_Directionality = All would be played out of the front speakers by the Chime Generator / Chime Audio Source.

Error Handling if chime requests that are not an allowable combination:

If the chime generator receives 2 chime requests with the same directionality (ie Chime\_Directionality & Chime\_Directionality2 both equal Front, both equal to Rear, or both equal to All) then the Chime Generator / Chime Audio Source shall follow the request and play both chimes out of the same speakers with the same directionality.

### 2.1.3.14 ALERT-SR-REQ-014750/C-Chime Audio Attenuation (TcSE ROIN-40370-1)

While chime(s) are active the Cluster shall tell the Chime Audio Source if the infotainment audio will be attenuated, muted, or if there will be no effect on non-chime infotainment audio via the 'IPC\_Infotainment.St(): Attn\_Info\_Audio' signal.

### 2.1.3.15 ALERT-SR-REQ-014752/C- Chime\_Active signal from Chime Generator (TcSE ROIN-40394-3)

The Chime Generator shall indicate what chime it is playing to the chime components (ex. Cluster, Chime Audio Source) via the '\_Chime\_Active = Chime\_X' signal. This event-periodic signal will be updated on event when the Chime Generator starts or stops playing a chime.

When there is no chime being played then the '\_Chime\_Active' signal shall equal 'No\_Chime'. Note: this does not mean that in between chime sounds (Off\_Time\_Btwn\_Chime) that 'No\_Chime' is set.



When the Chime\_Occurance signal is set to a fixed repetition rate the Chime Generator shall not set the \_Chime\_Active signal equal to "No\_Chime" until the last repetition and chime sound is finished playing. Until the last chime sound is played the \_Chime\_Active shall be set to the current chime.

#### 2.1.3.16 ALERT-SR-REQ-014753/D-Chime\_Occurance signal (TcSE ROIN-40395-1)

The Cluster shall tell the Chime Generator the Occurrence of the chime to be played using the Chime\_Occurance signal.

-- If Chime\_Occurance is set to 1, 2, 3... repetitions then the chime shall be played 1, 2, 3... times respectively and then the chime shall end (unless repetition ended early by the Cluster with IPC\_Chime : Chime = No\_Chime). The Cluster shall know that the chime has ended by looking at what the Chime Generator signal 'XXX\_Chime\_Active' is set equal to. If it's set to 'No\_Chime' then the chime has ended.

-- If Chime\_Occurance is set to 'Continuous' then the Chime shall play as long as the Chime\_X signal in 'IPC\_Chime : Chime = Chime\_X' doesn't change.

- When the chime is turned off (IPC\_Chime : Chime = No\_Chime) then the currently playing chime will end. It shall either finish playing the chime or immediately end depending on the Chime\_Time\_Criticality signal state.

#### 2.1.3.17 ALERTv2-SR-REQ-413977/A-Chimes / Prompts / Beep Prioritization

The Chime Audio Source shall be capable of mixing a chime and prompt together at the same time. If a beep event occurs while both a chime and prompt are active then the chime audio source shall mix in the beep if it is capable otherwise the beep shall not be heard. A chime has higher priority than a beep.

The Prompt Generator can monitor the IPC\_Infotainment : Attn\_Info\_Audio / New\_Attn\_Event signal to see if the prompt is muted or attenuated. This could be used by the prompt generator to avoid prompts seeming to be starting mid-sentence to the user because of a muting event caused by a chime.

#### 2.1.3.18 ALERT-SR-REQ-478177/A-Chime ID Assignments - variant 2 (Phoenix)

Chime_X	Chime Sound ID
Chime_0	No Chime
Chime_1	Turn Signal (Tic)
Chime_2	Turn Signal (Toc)
Chime_3	1.0 Second Chime
Chime_4	0.5 Second Chime
Chime_5	0.25 Sec Chime
Chime_6	1 Sec Tone (1KHz Alert)
Chime_7	0.1 Sec Chime
Chime_8	DNA Chime B (Soft Warning)
Chime_9	DNA Chime C (Hard Warning)
Chime_10	DNA Chime D (Non-Critical Alert) - Info
Chime_11	DNA "B" shortened to 0.5 sec
Chime_12	Perimeter Warn. Chime A
Chime_13	Perimeter Warn. Chime B
Chime_14	Perimeter Warn. Chime C
Chime_15	Cross-Traffic Alert (CTA)
Chime_16	Forward Park Aid
Chime_17	Reverse Park Aid
Chime_18	Not currently used
Chime_19	Not currently used
Chime_20	Not currently used
Chime_21	Not currently used
Chime_22	ACC-High, and FCW (1 cycle is 4 bursts repeated 3 times)
Chime_23	Lane Departure Warning (LDW)
Chime_24	Chime Message Initiated Push Button/ Touch Screen
Chime_25	Beltminder A



Chime_26	Beltminder B
Chime_27	RPA Continuous
Chime_28	FPA Continuous
Chime_29	Not currently used
Chime_30	300 msec Chime
Chime_31	Not currently used
Chime_32	Not currently used
Chime_33	Not currently used
Chime_34	Not currently used
Chime_35	Not currently used
Chime_36	Not currently used
Chime_37	Not currently used
Chime_38	Not currently used
Chime_39	Not currently used
Chime_40	Not currently used
Chime_41	Not currently used
Chime_42	Not currently used
Chime_43	Not currently used
Chime_44	Not currently used
Chime_45	Lane Departure Warning – for non-haptic systems
Chime_46	Not currently used
Chime_47	Hands on Wheel 1 - Highway Assist
Chime_48	Hands on Wheel 2 – Highway Assist
Chime_49	Hands on Wheel 3 – Highway Assist
Chime_50	BEV ON Chime
Chime_51	BEV OFF Chime
Chime_52	Drive Mode Transition
Chime_53	Not currently used
Chime_54	Not currently used
Chime_55	Not currently used
Chime_56	Not currently used
Chime_57	Not currently used
Chime_58	Not currently used
Chime_59	Not currently used
Chime_60	Not currently used
Chime_61	Not currently used
Chime_62	Beep Message Initiated (Push Button / Touch Screen)
Chime_63	Not currently used
Chime_64	Not currently used
Chime_65	Not currently used
Chime_66	Not currently used
Chime_67	Not currently used
Chime_68	Not currently used
Chime_69	Not currently used
Chime_70	Not currently used
Chime_71	Not currently used
Chime_72	Not currently used
Chime_73	Not currently used
Chime_74	Not currently used
Chime_75	Not currently used



Note: Chime\_X are referred to as Chime Sound in a number of requirements

#### 2.1.3.19 ALERT-SR-REQ-237862/D-Chime\_ID signal

Note throughout this Chime\_ID signal requirement when the SPSS specifies what to do for the ChimeID\_No\_Rq / ChimeID\_No\_Stat signals the same requirements apply to Chime2ID\_No\_Rq / Chime2ID\_No\_Stat signals.

- IPC\_Chime(ChimeID\_No\_Rq) corresponds to AHU\_Alert(ChimeID\_No\_Stat)
- IPC\_Chime2(Chime2ID\_No\_Rq) corresponds to AHU\_Alert(Chime2ID\_No\_Stat)

Note: The Chime\_ID signals were added per the Cluster Ford team since they are needed to identify chime active responses back from the AHU to make sure the Cluster Chime Request and AHU chime active responses (ie AHU\_Chime\_Active) were always talking about the same chime (chime = all the signals in IPC\_Chime/IPC\_Chime2). This was identified as a potential Cluster issue for multiple chime requests where Chime = Chime\_X for a sound which could be used for multiple chimes.

- This Chime ID requirement needs to be supported when Adjustable Chimes is supported. Once this requirement is supported in the AHU software then the Chime ID signal shall continue to be supported by the AHU regardless whether adjustable chimes is supported or not.

#### Cluster Chime Client:

The Cluster / Chime Client shall set the signal ChimeID\_No\_Rq equal to an ID value to uniquely identify all chime requests to the infotainment system (see chime arbitrator specs for details on implementing this in the Cluster).

Once the Cluster sets ChimeID\_No\_Rq equal to an ID value for a chime request it shall hold this value for the entire time this chime is active in the IPC\_Chime chime request message.

- Note: signal OFF\_Time\_Btw\_Chime is allowed to change and still be considered the same chime

The Cluster shall be backwards compatible with AHU modules that don't support the Chime\_ID signals (sourced to earlier versions of the Alerts SPSS). If not supported the ChimeID\_No\_Stat from the AHU should always be set to 0x0 Inactive.

Some chime sequence diagrams were updated to show the ChimeID\_No\_Rq and ChimeID\_No\_Stat signals. Reference those sequence diagrams for examples of those signals being used.

#### AHU Chime Generator:

The AHU Chime Generator shall set the signal ChimeID\_No\_Stat equal to the value in the Clusters ChimeID\_No\_Rq signal when playing that particular chime (IPC\_Chime(Chime = Chime\_X; ChimeID\_No\_Rq = Y...)). The AHU shall continue to send the same ChimeID\_No\_Stat value from the Cluster ChimeID\_No\_Rq chime request signal until the AHU is done playing the chime (either AHU finishes playing or the Cluster ends the chime). The AHU shall never change the ChimeID\_No\_Stat signal while playing a chime regardless what the Cluster is sending in its ChimeID\_No\_Rq signal.

When the AHU sets AHU\_Chime\_Active = No\_Chime the ChimeID\_No\_Stat signal shall be set to 0x1 (ID used for No\_Chime).

Some chime sequence diagrams were updated to show the ChimeID\_No\_Rq and ChimeID\_No\_Stat signals. Reference those sequence diagrams for examples of those signals being used.

In the case where Cluster doesn't support Chime ID signal but the AHU does then the AHU shall just repeat whatever is in the ChimeID\_No\_Rq signal. The ChimeID\_No\_Rq signal does not have any effect on the AHU chime audio itself so the AHU is backwards compatible with Clusters that don't support Chime ID signals..

- If the Cluster keeps ChimeID\_No\_Rq = 0x0 Inactive while chimes is active (because it doesn't support chimes) then the AHU shall set ChimeID\_No\_Stat = 0x0 Inactive.
  - Exception: If simpler on the AHU software when there is no chime the AHU could still set ChimeID\_No\_Stat = 0x01 ID for No Chime.

#### Example1 (AHU ends chime when finished playing it):

1. Cluster sends IPC\_Chime (Chime = Chime\_X; ChimeID\_No\_Rq = Z; Chime\_Occurrence = 5 repetitions;...)
2. AHU plays the chime and sends AHU\_Alert(AHU\_Chime\_Active = Chime\_X; ChimeID\_No\_Stat = Z) while playing the chime



3. AHU finishes the 5<sup>th</sup> repetition and sends AHU\_Alert(AHU\_Chime\_Active = No\_Chime; ChimeID\_No\_Stat = 0x01 ID for AHU No Chime).

Example 2 (Cluster ends a chime before the AHU is finished playing it):

1. Cluster sends IPC\_Chime (Chime = Chime\_X; ChimeID\_No\_Rq = Z; Chime\_Occurence = 5 repetitions;...)
2. AHU starts playing the chime and sends AHU\_Alert(AHU\_Chime\_Active = Chime\_X; ChimeID\_No\_Stat = Z)
3. Cluster ends the chime before the AHU finishes playing the 2<sup>nd</sup> repetition chime sound and sends IPC\_Chime(Chime = No\_Chime; ChimeID\_No\_Rq = don't care for AHU; Chime\_Time\_Criticality = Criticality\_Low)
4. AHU sends AHU\_Alert(AHU\_Chime\_Active = Chime\_X; ChimeID\_No\_Stat = Z) until the second repetition Chime\_X sound is complete
5. When AHU finishes the 2<sup>nd</sup> repetition it sends AHU\_Alert(AHU\_Chime\_Active = No\_Chime; ChimeID\_No\_Stat = 0x01 ID for AHU No Chime)

DSP AMP chime audio source:

No impact to the DSP AMP

#### 2.1.3.20 ALERT-SR-REQ-052682/M-Chime Requests while the infotainment system is muted during a cold crank

This requirement does not apply for BEV vehicles (Battery Electric Vehicles). For Battery Electric Vehicles only (hybrids not included) there shall be no muting for chimes when the Crank CAN signals are received showing a crank event as defined in "STMGNTv2-FUN-014669-Crank, Front System ON". For gas vehicles and hybrids what is below in this requirement applies. BEV vehicles don't support cold cranks.

- Note: talk with Ford D&R engineer regarding what architecture to implement this on. Needs to be a coordinated change for both the AHU and DSP AMP if support on the FNV2 architecture. For FNV3 all modules need to support.

During a cold crank event infotainment chimes are muted by the infotainment system. If the Cluster needs to have a chime played during a cold crank while the infotainment system is muted, then the Cluster would have to play the chime through the Cluster.

After a crank event ends the Chime Generator shall unmute and be capable of playing the chimes no later than ~~4000 msec~~ **500 msec** after the crank event ends.

- The Crank event ending for both the AHU, DSP AMP, DSP AMPv2, AHUv2 is defined in requirements "STMGNTv2-GFUN-202153-1-Crank, Front System ON" for CGEA 1.3 and "STMGNT-FUN-014666-Crank, Front System ON" for CGEA 1.2 / C1MCA.
- Reference the EMC specification(s) for the worst case cold crank voltage profile that needs to be supported (ex EMC requirement CI-230).

After a crank event ends but before the Chime Generator has un-muted if the chime generator receives a chime request then:

- the Chime Generator shall still respond to IPC\_Chime/IPC\_Chime2 chime request messages and behave as if it is playing the chime (ex chime generator responds with AHU\_Alert : AHU\_Chime\_Active = Chime\_X)
- as soon as the Chime Generator is un-muted the chime will be playing as if the chime began playing with the initial chime request (exception seat belt chime). For example, if the chime had 3 repetitions then by the time the Chime Generator is un-muted it may be on the 2<sup>nd</sup> or 3<sup>rd</sup> repetition.

#### Exception to Chime Generator requirement above - Seat Belt Chime:

After a crank event ends and the Chime Generator has not yet un-muted if the chime generator had received a Seat Belt Chime request from the Cluster while the network signal said crank (if message was not lost with crank voltage dip) or after crank ends but before the AHU / DSP AMP un-muted then (ie IPC\_Chime /IPC\_Chime2 = "Seat Belt Chime") then:

1. the chime generator shall update the AHU\_Chime\_Active/AHU\_Chime\_Active2 signal to Seatbelt Chime when it received the IPC\_Chime request (follow sequence diagram for when DSP AMP present), and
  - AHU Only (No DSP AMP):
    - then the chime generator delays playing the seat belt chime until after the chime generator is un-muted as long as:
      - i. when the unmute first occurs the Cluster is still sending the seat belt chime request message (IPC\_Chime = "Seat Belt Chime")
      - ii. and the Cluster has not already ended the chime (ie IPC\_Chime : Chime = No\_Chime).





The Chime Generator shall start playing the seat belt chime no later than 20 msec after the unmute occurs.

- AHU with DSP AMP combination or Phoenix Architecture (APIM PDC):
  - then the chime generator delays playing the seat belt chime until ~~950~~ 500 msec +/- 10 msec after the crank event ends as defined in Station Management SPSS crank requirements (STMGNT-FUN-014666, STMGNT-FUN-202153) as long as:
    - when the unmute first occurs the Cluster is still sending the seat belt chime request message (IPC\_Chime = "Seat Belt Chime")
    - and the Cluster has not already ended the chime (ie IPC\_Chime : Chime = No\_Chime).

The IPC\_Chime / IPC\_Chime2 seat belt chime is active when the following signals are set:

Ford Seat Belt Chime:

IPC\_Chime/IPC\_Chime2 (Chime = 0x0B Chime\_10, Chime\_Vol\_Levl = 0x0 100% stored volume; Chime\_Directionality = 0x2 Front; Chime\_Time\_Criticality = 0x1 Criticality\_High; OFF\_Time\_Btwn\_Chime = 0x1 Continuously; Chime\_Occurance = 0x4 4 repetitions)

Lincoln Seat Belt Chime:

IPC\_Chime/IPC\_Chime2 (Chime = 0x15 Chime\_20, Chime\_Vol\_Levl = 0x0 100% stored volume; Chime\_Directionality = 0x2 Front; Chime\_Time\_Criticality = 0x1 Criticality\_High; OFF\_Time\_Btwn\_Chime = 0x1 Continuously; Chime\_Occurance = 0x4 4 repetitions)

Example – Seat belt chime request after a crank event ended but before the AHU is unmuted (No DSP AMP present):

1. Crank event ends
2. The Cluster sends IPC\_Chime (Chime = 0x0B Chime\_10, Chime\_Vol\_Levl = 0x0 100% stored volume; Chime\_Directionality = 0x2 Front; Chime\_Time\_Criticality = 0x1 Criticality\_High; OFF\_Time\_Btwn\_Chime = 0x1 Continuously; Chime\_Occurance = 0x4 4 repetitions)
3. The AHU responds within 75 msec with AHU\_Alert.st (AHU\_Chime\_Active = 0xA Chime\_10)
4. 300 msec (could be any value under 500 msec) after the crank event ended the AHU unmutes and starts playing the first seat belt chime sound and plays all 4 repetitions (unless Cluster ends the chime before finished).

2.1.3.21 ALERT-SR-REQ-413978/A-AHU variant 2 and DSP AMP variant 2 chime functionality

The AHUv2 (AHU variant 2) and DSP AMPv2 do not support IPC\_Chime / IPC\_Chime2 chime requests since the AHUv2 and DSP AMPv2 are not the Chime Generator or Chime Audio Source.

The Chime Generator / Chime Audio Source for the A2B chime outputs to the AHUv2 and DSP AMPv2 is responsible for muting and unmuting between chime events.

The AHUv2 and DSP AMPv2 do support chime diagnostics as defined in the IDS and applicable requirements in this chime spec for the Chime\_Supported signal.

The AHUv2 and DSP AMPv2 are mutually exclusive with the AHU and DSP AMP. In this chime specification when it says AHU or DSP AMP it does not apply to the AHUv2 or DSP AMPv2.

Note: see requirement "VOLv2-REQ-412367-Module Deployment and Audio Routing" for chime audio routing.

2.1.3.22 ALERT-SR-REQ-361909/B-VMCU Chime Generator

The VMCU back-up / early chime generator shall be capable of playing all chimes out of the back-up speaker(s).

- Note: this does not require the audio quality to be as good as when through the CCPU

The VMCU back-up / early chime generator shall be responsible for playing the chimes when the CCPU cannot play the chimes.

Note: See "Alert-REQ-413987-Chime Error States / Fault handling – variant 3" for details of fault handling and using the PDC\_Chime\_Supported signal regarding the VMCU being the back-up chime generator



## 2.1.4 Chime Error Management Requirements

### 2.1.4.1 ALERTv3-SR-REQ-413982/A-Loss of communication with Chime module - variant 3

The Cluster VMCU shall monitor the Chime Generator CCPU internal to APIM PDC to make sure it can play chimes. The VMCU shall monitor the CCPU and if the CCPU is no longer detected (ex rebooted/reset) while chimes are through the infotainment system then the VMCU shall play the chimes through the back-up speaker and set the signal 'IPC\_Infotainment.St() : Chime\_Source = Cluster' and 'Power\_Up\_Chime\_Modules = Inactive'.

The Cluster VMCU shall monitor the PDC\_Alert heartbeat periodic message for determining if the Chime Generator message is available. If the VMCU loses communication with the CCPU Chime Generators PDC\_Alert message for more than 5 seconds then the VMCU back-up generator shall become the Chime Generator and Chime Audio Source and shall set the signal 'IPC\_Infotainment.St() : Chime\_Source = Cluster' and 'Power\_Up\_Chime\_Modules = Inactive'.

- Note: this is specific to PDC\_Alert heartbeat periodic message missing. For other error conditions if the VMCU knows there is an issue and no chimes can be generated by the CCPU then the VCMU can be the back-up chime generator and does not need to wait the 5 seconds.

If the Chime Generator loses communication with the AHUv2 or AHUv2 and DSP AMPv2 (when DSP AMP present) for more than 5 seconds, then the Chime Generator shall set Chime\_Supported = Not\_Supported so the Cluster (VMCU) plays the chimes through the back-up speaker.

### 2.1.4.2 ALERTv3-SR-REQ-413987/B-Chime Error States / Fault handling - variant 3

If the Chime Generator (ie CCPU) sets its signal 'Chime\_Supported' equal to 'Not\_Supported' for a particular configuration then the Cluster (ie VMCU) shall become the Chime Generator and Chime Audio Source and sets the 'IPC\_Infotainment.St() : Chime\_Source = Cluster' and Power\_Up\_Chime\_Modules = Inactive. When the Cluster is the Chime Generator and Chime Audio Source it shall play the chimes out of the back-up speaker.

The Chime Generator (ie CCPU) shall monitor the AHUv2 AHU\_Chimes\_Supported and DSP AMPv2 DSP\_Chimes\_Supported network signals. If either the AHU\_Chimes\_Support signal or DSP\_Chimes\_Supported signal equals Not\_Supported then the CCPU shall I set its PDC\_Chimes\_Supported signal = Not\_Supported.

The Chime Generator (ie CCPU) shall set its PDC\_Chimes\_Supported signal equal Inactive, instead of Supported, if either the AHUv2 AHU\_Chimes\_Supported or DSP AMPv2 (when present) DSP\_Chimes\_Supported signals are set to Inactive.

- This is applicable as long as no Chime\_Supported signals equal Not\_Supported

The Chime components are responsible for setting "\_Chimes\_Supported = Not\_Supported" (ex APIM PDC CCPU, AHUv2, DSP AMPv2) whenever they are no longer able to produce chimes because of a fault condition, otherwise their signal shall be set to 'Supported' if able to produce chimes with no fault conditions (see Infotainment Diagnostic spec for details).

- A normal crank event is not considered a fault condition. A normal crank event would not cause the \_Chime\_Supported signal to change. For example, if Chime\_Supported = Supported and a normal crank event occurs the \_Chime\_Supported value would remain equal to Supported and would not change because of the crank event (ie would never change to Inactive or Not\_Supported).
- A chime FMEA shall be performed on all the chime components to verify that fault conditions are detected and \_Chimes\_Supported is set to Not\_Supported when necessary.

Any infotainment component that could prevent chimes from being played through the infotainment system needs to have a "XXX\_Chimes\_Supported" signal so chimes can be transferred to the chime back-up speaker when needed for an error condition.

Some fault conditions (while not limited to these) that would result in the AHUv2, DSP AMP variant 2 \_Chimes\_Supported signal being set to Not\_Supported: 1) Short/open circuit to any of the chime speakers 2) short/open circuit on the audio signals to the chime components 3) low voltage preventing the chime components from producing audio (not applicable to crank but if stuck at a low voltage) 4) A2B link error / A2B loss of communication (see A2B SPSS for details)...

When the Audio Enable line is keeping the Audio Muted for more than 5 seconds then the amp being muted shall set its chime signal 'XXX\_Chime\_Supported' equal to 'Not\_Supported' (example DSP AMPv2).



The Chime Audio Components (ex. APIM PDC CCPU, AHUv2, DSP AMPv2...) shall set their \_Chime\_Supported signal to Supported or Not\_Supported within 2000 msec of the start of Chime Diagnostics. Reference the IDS (infotainment diagnostic spec) for chime diagnostics.

If \_Chime\_Supported = Supported and then Power\_Up\_Chime\_Modules = transitions from Active to Inactive and the module powers down and is no longer capable of producing chimes then the \_Chime\_Supported signal shall equal "inactive".

- Note: the start of chime diagnostics requires \_Chime\_Supported signal to equal Inactive (see IDS for details)

~~If the Chime Generator (ie CCPU) doesn't receive the 1. AHUv2\_AHU\_Chime\_Supported signal (no DSP AMPv2 present) or 2. AHUv2\_AHU\_Chime\_Supported and DSP AMPv2\_DSP\_Chimes\_Supported signals (if DSP AMP present) equal to Supported within 3000 msec after the start of chime diagnostics it shall treat 'Inactive' the same as 'Not\_Supported'.~~

~~If the Chime Client (Cluster) receives 'PDC\_Chimes\_Supported = Inactive' from the CCPU 5000 msec after Power\_Up\_Chime\_Modules transitions from Inactive to Active (or at network bus start up might just go directly to active) then it shall treat 'Inactive' the same as 'Not\_Supported'.~~

Note: The IPC\_Infotainment : Chime\_Source signal has no effect on the \_Chime\_Supported signal.

A periodic (if applicable) IPC\_Chime/IPC\_Chime2 network message is not considered a new chime event for the Chime Audio Components (ex Chime Generator, Chime Audio Source).

The user adjusted Bass, Treble, Balance, Mid-Range, Fade settings shall not affect chimes.

- For example if the BTMBF setting did effect chimes the user could fade to front speakers and a chime that is supposed to be played out of the rear speakers only would not be heard.

If there are not event-periodic (1000 msec) chime signals (like CAN set-up) between the VMCU and CCPU then new error handling needs to be developed in case of a missing IPC\_Chime, IPC\_Chime2, IPC\_Infotainment and PDC\_Alert signal.

- For example (without event-periodic signals, but only event signals),
  1. if there is a continuous chime playing, and then:
  2. the IPC\_Chime : No\_Chime is sent to the CCPU to end the chime but the signal is lost (ex long cold crank in cold weather for a few seconds), then if the IPC\_Chime signal is no longer periodic then the CCPU could potentially play the chime indefinitely for that ignition cycle.
    - This won't be an issue if the signals are made event-periodic like CAN or if some other error handling strategy is developed.

#### 2.1.4.3 ALERT-SR-REQ-014760/C- Chime\_Supported signal changes to Supported from Not\_Supported (TcSE ROIN-40444-2)

If the Chime Generator set its signal 'XXX\_Chimes\_Supported' equal to 'Supported' after previously being set to 'Not\_Supported' in the same ignition state then the Cluster shall not set the infotainment system as the chime source until the next ignition cycle. The 'IPC\_Infotainment.St() : Chime\_Source' shall remain equal to 'Cluster' for that ignition state.

#### 2.1.4.4 ALERTv2-SR-REQ-433163/A-Load Shed - variant 2

During an infotainment Load Shed event (HMIAudioMode = Load Shed) the Cluster shall set the 'IPC\_Infotainment.St() : Chime\_Source' signal equal to 'Cluster' AND Power\_Up\_Chimes\_Modules = Inactive ~~for that ignition cycle~~ and the Cluster shall become the Chime Generator and Chime Audio Source.

Note: see "[PWRMAN-GREQ-014509-Infotainment Components Load Shed State requirements](#)" for when an infotainment load shed event is active

#### 2.1.4.5 PWRMAN-SR-REQ-014509/I-Infotainment Components Load Shed State requirements (TcSE ROIN-66172-3)

Unless otherwise noted the infotainment components shall transition to their Standby or Sleep Load Shed low power state when the signal HMIAudioMode == Load Shed.





In the Standby Load Shed low power state non-essential component functions shall be turned OFF (ex. active pre-fetch). Basic standby operations will still be followed such as supporting the Network bus and any regulatory requirements.

Note: There may also be applicable Climate Control load shed requirements for modules that support Climate Control functionality. See Climate specifications for details.

#### Chimes and Load Shed:

- The infotainment components that support chimes (ex. AHU, DSP AMP, AAM, ANC...) during a transition to load shed from state where chimes are through the infotainment system shall wait until the Cluster transfers control of the chimes back to the Cluster (as defined in ALERT-REQ-014761-Load Shed) before entering their low power states.
- Since the infotainment components that support chimes have to wait for Cluster to transfer chime control back to the Cluster (ie IPC\_Infotainment : Chime\_Source = Cluster) during a load shed event before they no longer support chimes the infotainment components would have Chime\_Supported = Supported while supporting chimes.
- After chime control is transferred to the Cluster the infotainment components shall change Chime\_Supported = Not\_Supported while the load shed is active.

#### Chimes and Load shed – variant 3: (applies to Phoenix architecture)

- APIM PDC (Cluster and Chime Generator in one module): when HMIAudioMode = Load Shed the integrated Chime Client and Chime Generator shall use the VMCU back-up speaker and set Power\_Up\_Chime\_Modules = Inactive and Chime\_Source = Cluster.
- The PDC CCPU, AHUv2 and DSPv2 shall set their Chime\_Supported signals equal to ~~Not\_Supported~~ "Inactive" while a load shed event is active.
  - Note: Chime\_Supported set to Inactive allow the chimes to go back to the infotainment system when the load shed event ends (ex engine is running). Chime\_Supported = Not\_Supported would lock out chimes for that whole ignition cycle.

#### ~~Chimes and Load Shed – variant 2: (applies to SYNC 4.2)~~

- ~~SYNC 4.2 (AHU and Cluster in one module): when HMIAudioMode = Load Shed the integrated Chime Client and Chime Generator shall use the back-up speaker(s) and set Power\_Up\_Chime\_Modules = Inactive and Chime\_Source = Cluster (as defined in "Alertv2-REQ-372081-Load Shed – SYNC 4.2").~~

#### 2.1.4.6 ALERT-SR-REQ-014762/D-Response when chime signals from Chime Client set to inactive (TcSE ROIN-193435-1)

If any of the signals Chime, Chime\_Directionality, Chime\_Time\_Criticality, OFF\_Time\_Btw\_Chime, or Chime\_Occurance in the IPC\_Chime (1 or 2) messages are set to 'inactive' then the Chime Generator won't create a new chime sound (ex Chime\_8 DNA B) and the AHU\_Alert : AHU\_Chime\_Active signal will be set to No\_Chime to tell the Cluster that it is not playing the chime.

#### 2.1.4.7 PWRMAN-SR-REQ-014520/H-Transport Mode and CGEA Chimes (TcSE ROIN-40663-3)

Audio Chimes shall NOT be enabled through the Infotainment System during Transport Mode. The Cluster shall support Chimes during Transport Mode.

During a transition to Transport Mode from another LifeCycMde\_D\_Actl/CarMode state where the chimes are through the infotainment system the Cluster shall set the 'Chime\_Source' signal equal to 'Cluster' and Power\_Up\_Chime\_Modules = Inactive.

- ~~The infotainment components that support chimes shall wait until the Cluster transfers control of the chimes back to the Cluster with the 'Chime\_Source = Cluster' AND 'Power\_Up\_Chime\_Module = Inactive' before entering their transport mode low power states.~~
  - ~~Since the infotainment components that support chimes have to wait for Cluster to transfer chime control back to the Cluster during a transport mode event before they no longer support chimes the infotainment components would have Chime\_Supported = Supported while supporting chimes. After chime control is transferred to the Cluster the infotainment components can change Chime\_Supported = Not\_Supported while transport mode is active.~~
    - Legacy Infotainment components can continue follow the requirement above with strikethroughs (if already implemented) where they wait for the Cluster before setting Not\_Supported. New modules



(ex new AHU, DSP AMP) shall set Chime\_Supported = Not\_Supported when LifeCyclMde\_D\_Actl = Transport.

- For the Phoenix architecture the AHUv2 and DSP AMPv2 shall set their Chime\_Supported signals to Not\_Supported when LifeCycMde\_D\_Actl = Transport.

## 2.1.5 Sequence Diagrams

### Note:

1. In the sequence diagrams for the Phoenix architecture the Chime Generator and Chime Audio Source are the same module (ex APIM PDC) and the sequence diagrams do not include the AHUv2 and DSP AMPv2. Because of this the interface between the Chime Generator and Chime Audio Source can be ignored since this is all internal to the CCPU.
2. Any sequence diagrams that call out <CAN> should ignore the CAN label and treat the signals as logical signals not tied to specific network
3. Some sequence diagrams (ex on different alert channels in particular) still show the Cluster sending chime requests without waiting for the no chime from the AHU. For Phoenix architecture this does not apply, and the Cluster shall follow requirement "Alert-REQ-014741-Cluster Chime-Requests" which requires the Cluster to wait for the no chime from the Chime Generator.

### 2.1.5.1 ***ALERT-SD-REQ-014763/D-Cluster ending Continuous Chime event, or Fixed Repetition Chime event where Cluster ends chime before repetitions finished (TcSE ROIN-40403-4)***

#### **Pre-condition**

No Chimes are active

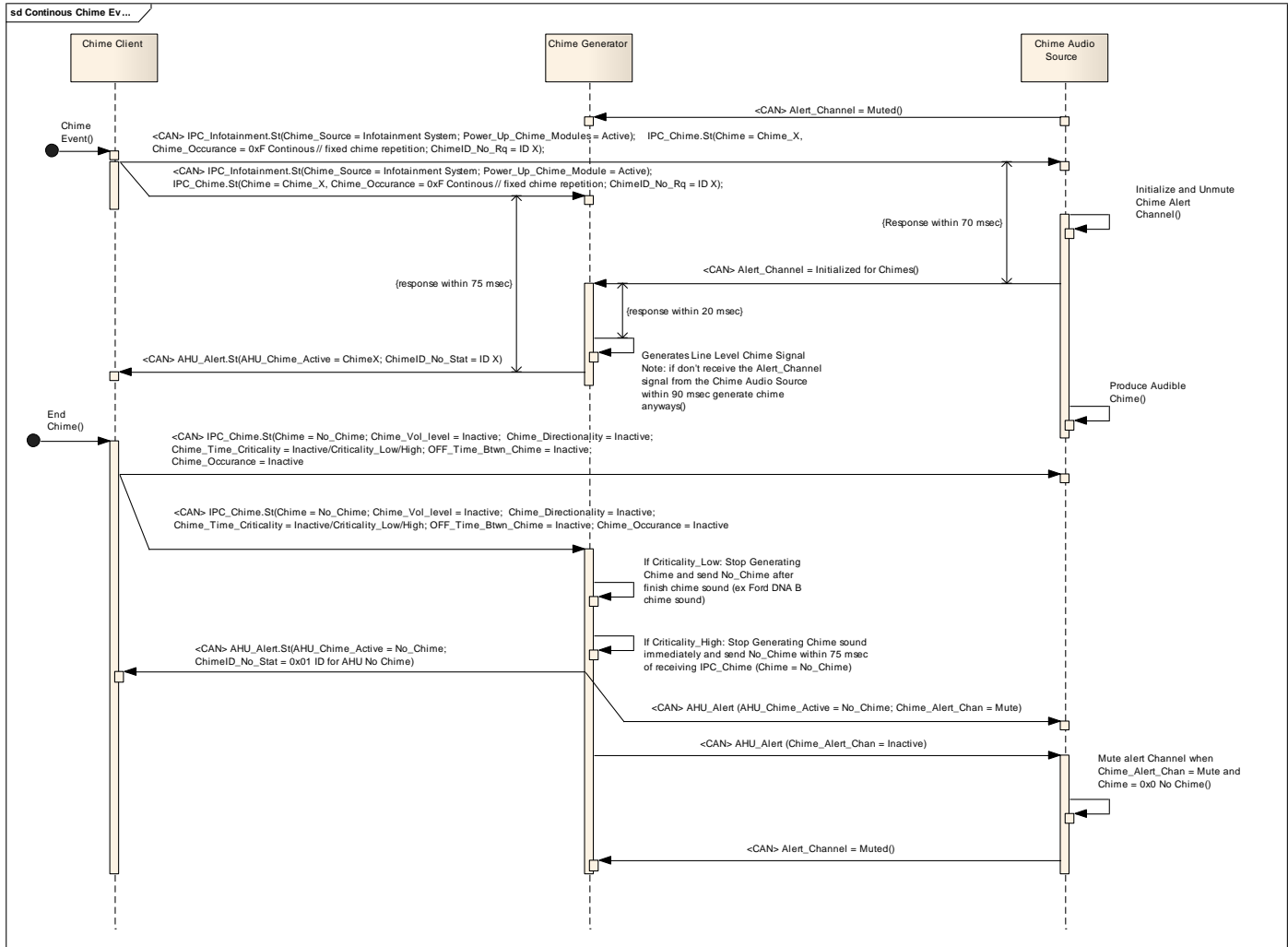
#### **Scenario**

Chime Client initiates a chime

#### **Post-condition**

Chime is ended by the Chime Client

### Sequence Diagram



#### 2.1.5.2 ALERT-SD-REQ-014764/B-Fixed Repetitive Chime Event (TcSE ROIN-167429-2)

##### Pre-condition

No Chimes are active

##### Scenario

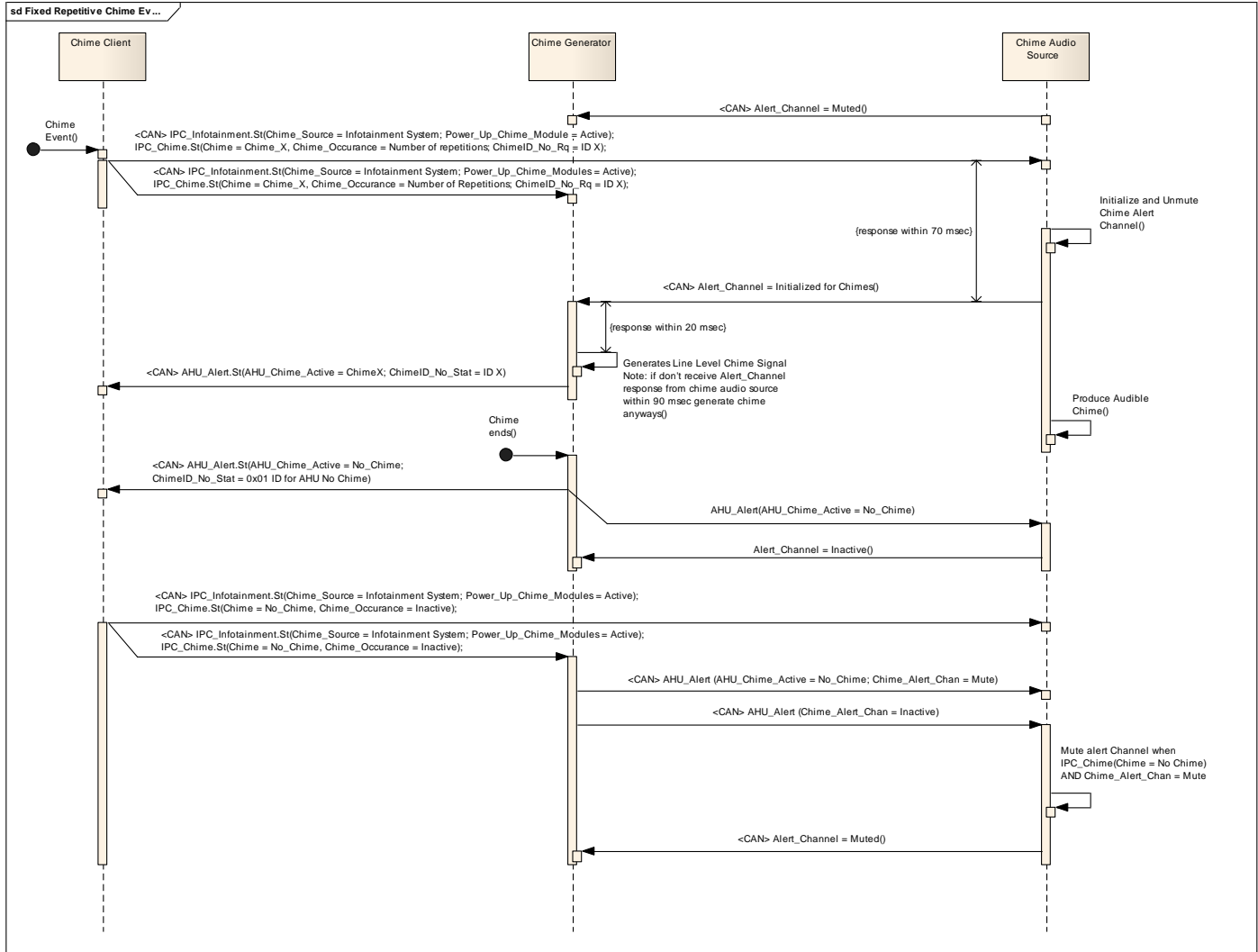
Chime Client initiates a Fixed Chime Event

##### Post-condition

Chime is completed



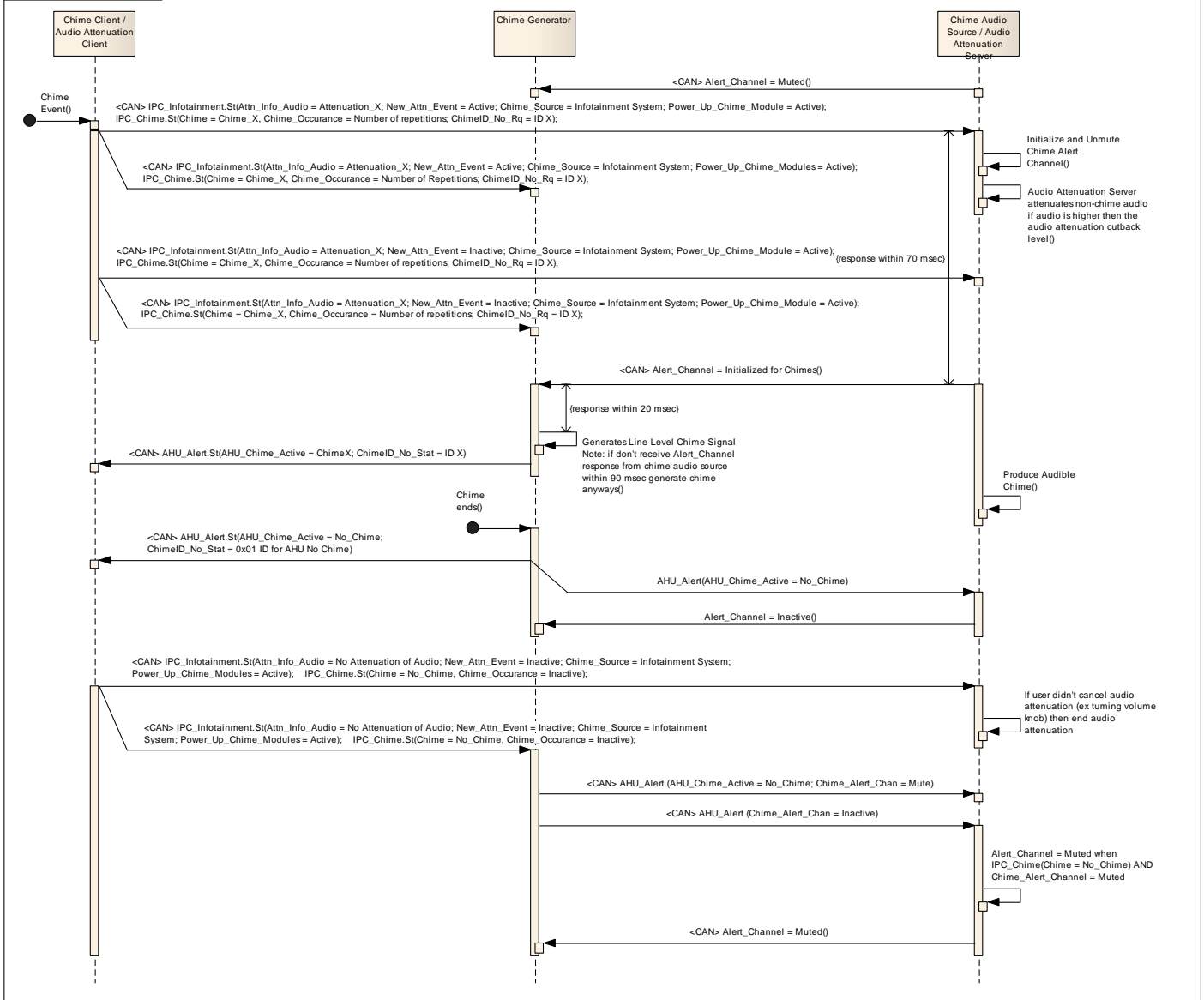
## Sequence Diagram



## Sequence Diagram

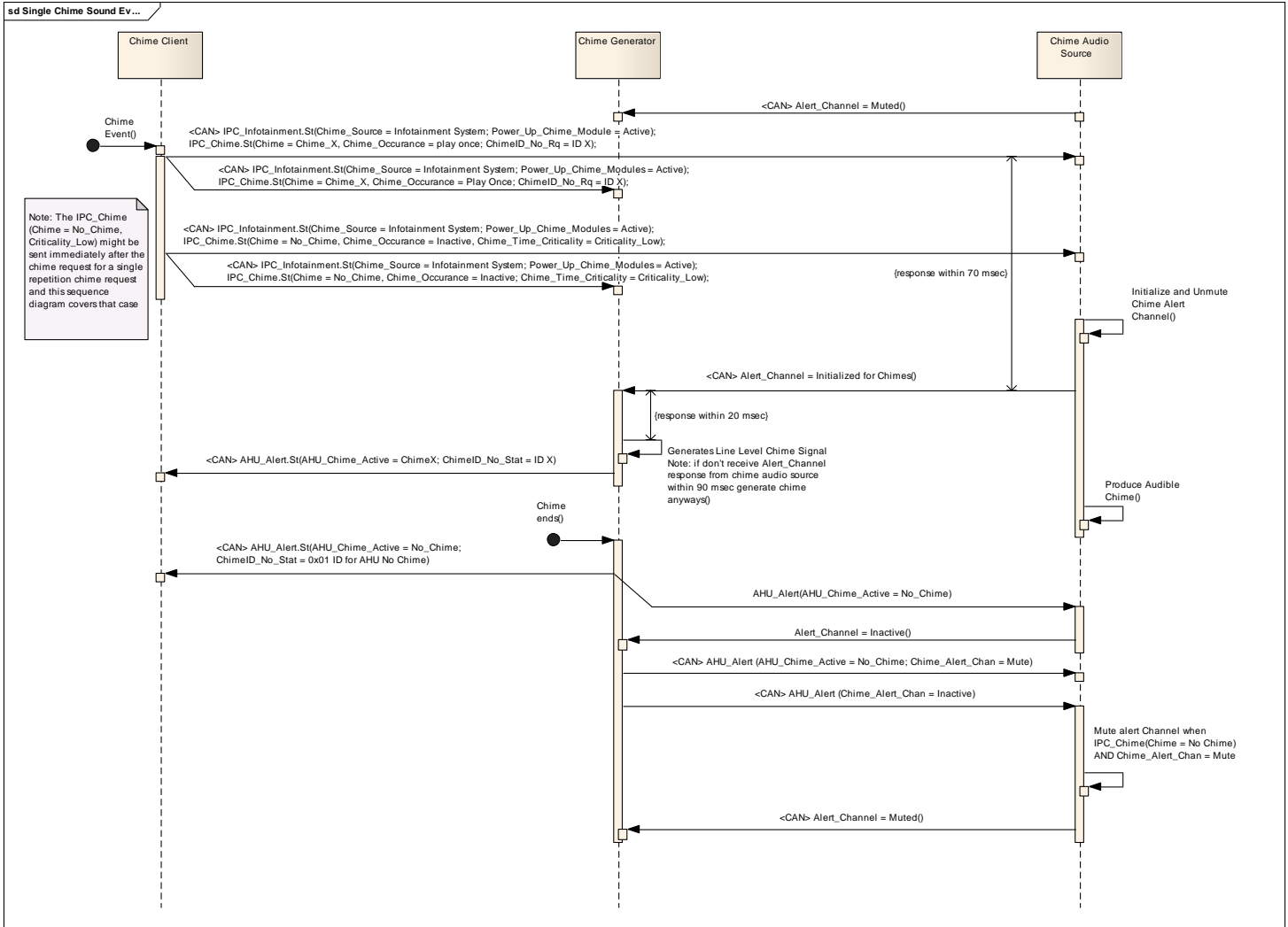


sd Audio Attenuation Chime Ev...





## Sequence Diagram

**2.1.5.3 ALERT-SD-REQ-014765/A-Low Priority Chime overwriting another Chime on the same Alert Channel (TcSE ROIN-200749-2)****Pre-condition**

Chime event 1 is active on Alert Channel 1

**Scenario**

Chime Client initiates a low criticality Chime event 2 on Alert Channel 1

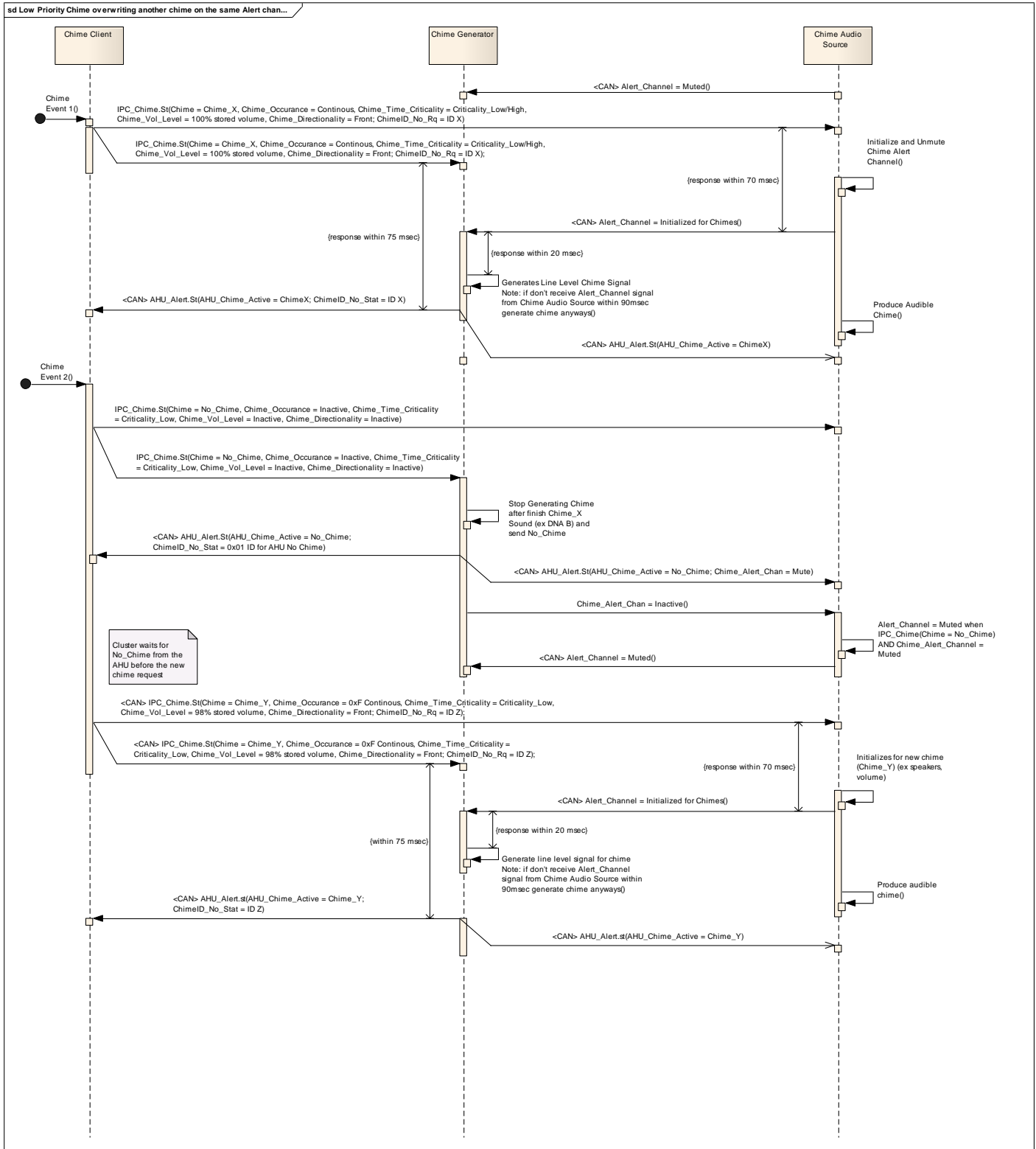
**Post-condition**

Chime event 2 is playing through the infotainment system on Alert Channel 1

## Sequence Diagram



sd Low Priority Chime overwriting another chime on the same Alert chan...



#### 2.1.5.4 ALERT-SD-REQ-014766/A-High Priority Chime overwriting another Chime on the same Alert Channel (TcSE ROIN-200760-3)

##### Pre-condition

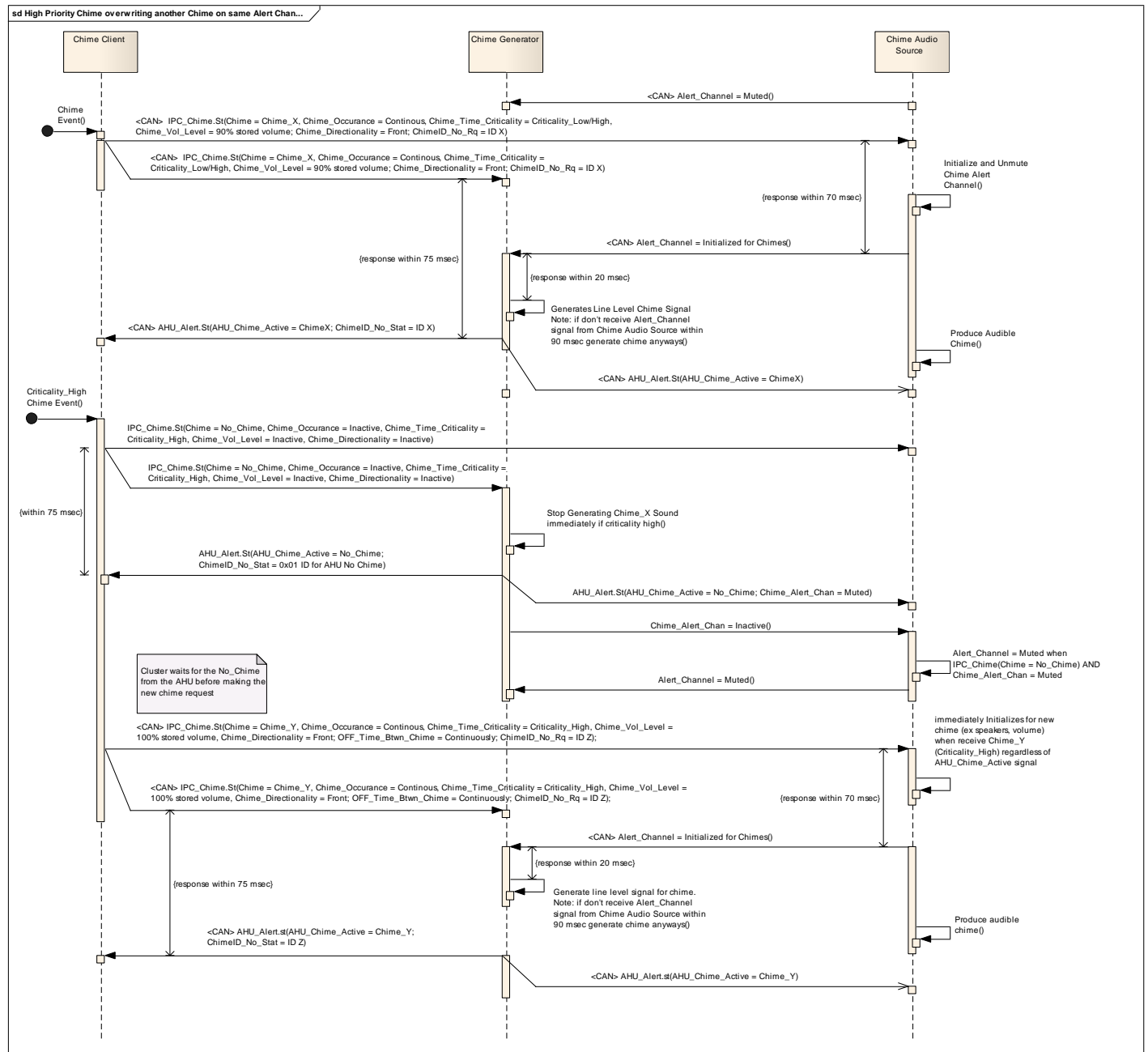
There is a chime active on Alert Channel 1

**Scenario**

Chime Client initiates a High Criticality Chime on Alert Channel 1

**Post-condition**

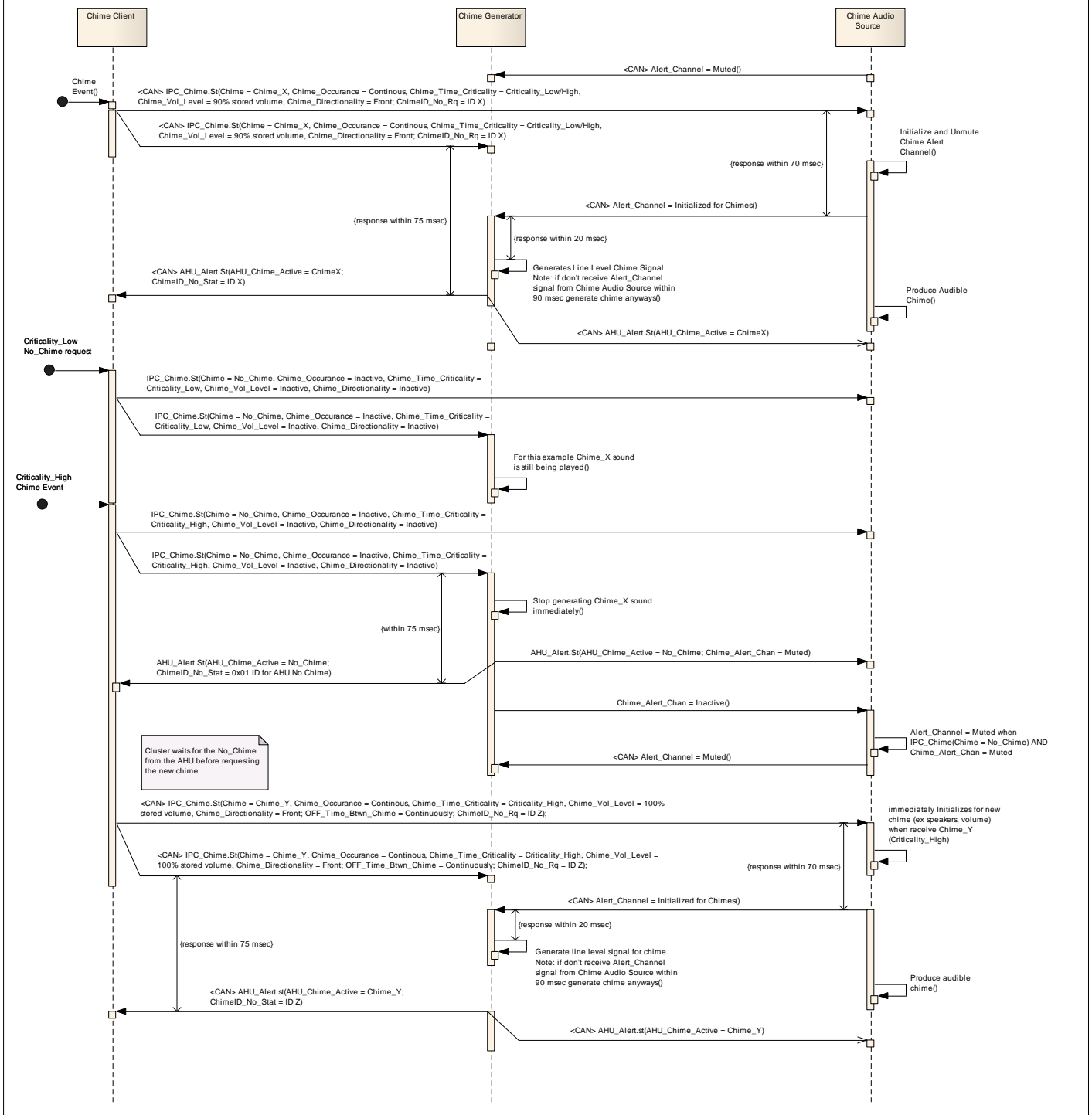
The new high priority chime is playing through the infotainment system on Alert Channel 1

**Sequence Diagram****Sequence Diagram**





sd ex 2 - High Priority Chime overwriting another Chime on same Alert Chan...



### 2.1.5.5 ALERT-SD-REQ-014767/A-Low Priority Chime - requesting a new chime on a different Alert channel with the same directionality (TcSE ROIN-285762-1)

#### Pre-condition

Chime event 1 is active on Alert Channel 1

#### Scenario

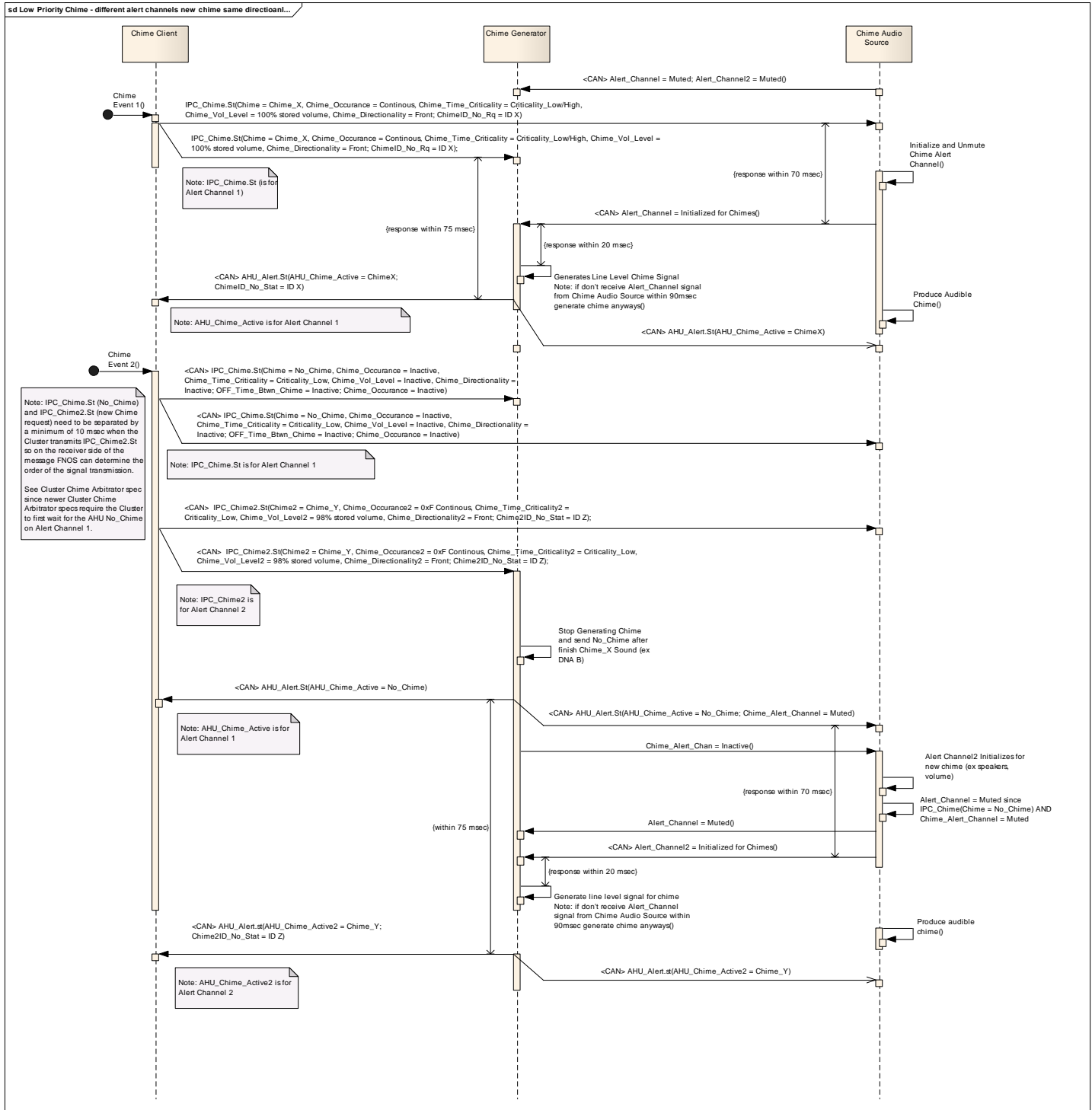
Chime Client initiates a new low criticality Chime event 2 on Alert Channel 2



## Post-condition

Chime event 2 is playing through the infotainment system on Alert Channel 2

## Sequence Diagram





### 2.1.5.6 ALERT-SD-REQ-014768/D-High Priority Chime - requesting a new chime on a different Alert channel with the same directionality (TcSE ROIN-285767-1)

#### Pre-condition

Chime event 1 is active on Alert Channel 1

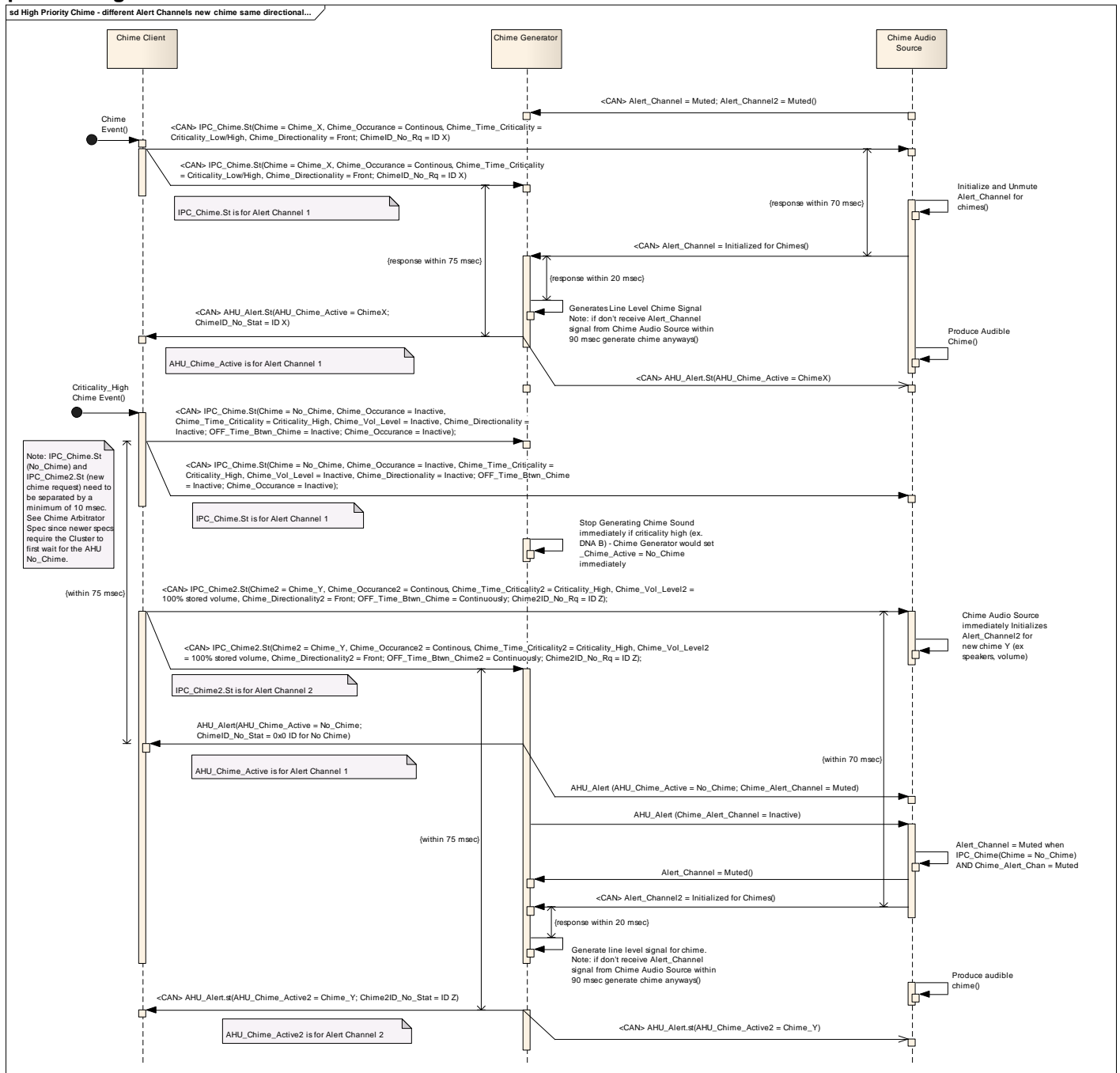
#### Scenario

Chime Client initiates a new high criticality Chime event 2 on Alert Channel 2

#### Post-condition

Chime event 2 is playing through the infotainment system on Alert Channel 2

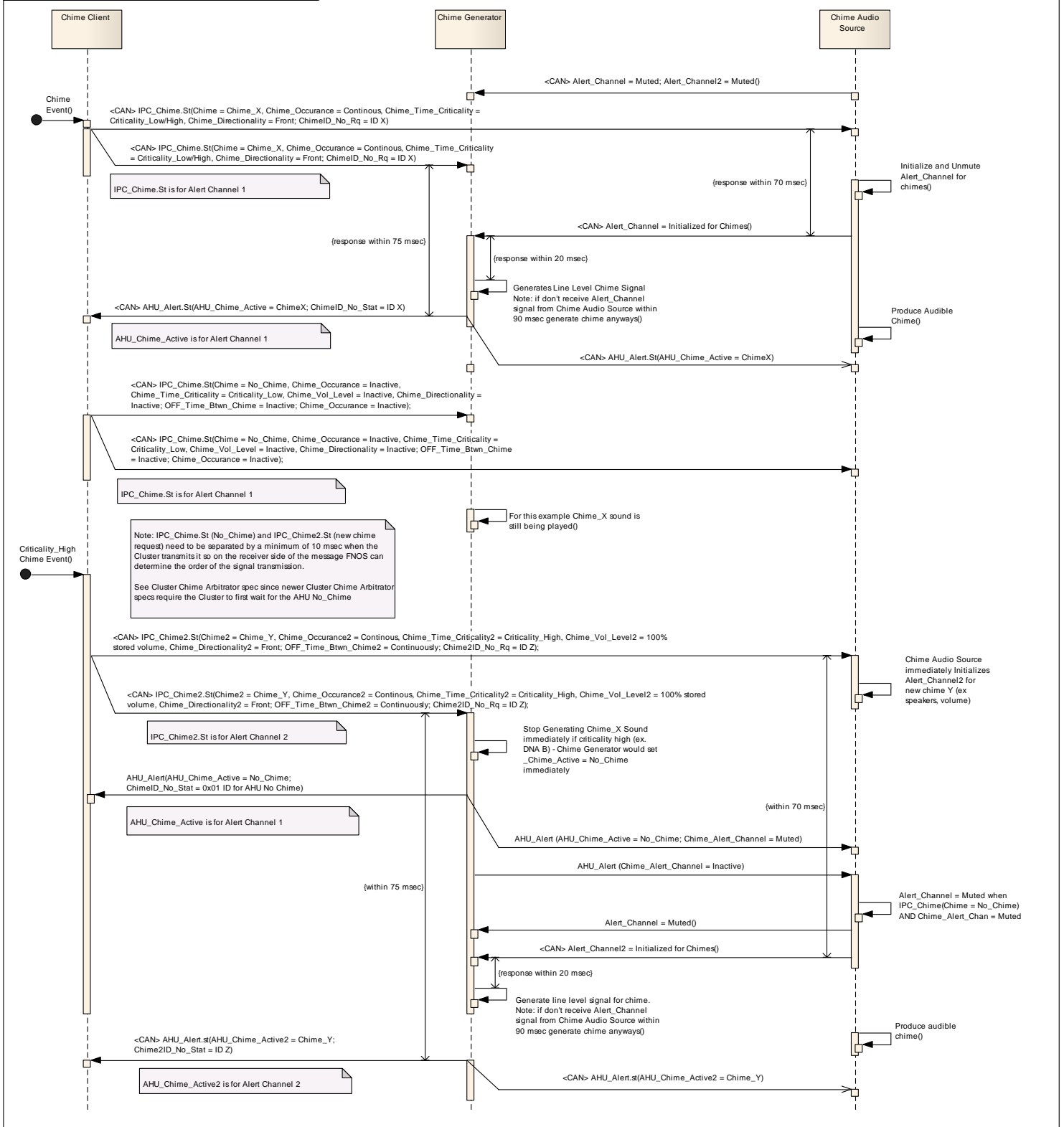
### Sequence Diagram





## Sequence Diagram

sd High Priority Chime 2 - different Alert Channels new chime same directional...





## 2.2 Adjustable Chimes Deployment

The table below shows how the logical classes may be mapped to physical modules for the Adjustable Chime feature. The table below covers the lead program.

At the time the specification was written the below table was the latest. If there are additional modules deployed to the class descriptions or the vehicle architecture changed since the spec was written and released, then the applicable implementation guide class description would cover those modules. If there is a conflict between the implementation guide and the table below the implementation guide takes precedent.

Logical Class	APIM PDC	APIM PDC VMCU	APIM PDC CCPU	DSP AMP variant 2	AHU variant 2
Chime Client (Chime Master)	x	x		N/A	N/A
Chime Generator	x		x		
Chime Audio Source	x		x		
Adjustable Chime Server	x	x			
Adjustable Chime Client	x		x		
Cluster	x	x			
AHU	x		x		
DSP AMP	N/A	N/A	N/A		

- X – represents functionality supported by that module
- Blank – represents functionality not supported by that module

Note: This is all internal to the APIM PDC (Phoenix Domain Controller) for adjustable chimes. When spec was written there were separate AHU and DSP AMP modules that were part of adjustable chimes, but with Phoenix it is all internal to the APIM PDC. This includes any references to the AHU chime generator AHU\_Chime\_Supproted signal to the Cluster which is all now logically internal to the APIM PDC (ie PDC\_Chimes\_Supported).

At the time this spec was written the VMCU and CCPU used the following signals to communicate the Adjustable chimes settings:

- RPA\_Adj\_Chime\_Vol\_MC\_Cmd
- FPA\_Adj\_Chime\_Vol\_MC\_Cmd
- RPA\_Adj\_Chime\_Vol\_MC\_St
- FPA\_Adj\_Chime\_Vol\_MC\_St

Reference the “Settings for Vehicle Interface Processor in Integrated Cluster” STSS for details.

## 2.3 ALERT-FUN-REQ-195348/D-Adjustable Chime Strategy

### 2.3.1 Use Cases

#### 2.3.1.1 ALERT-UC-REQ-195369/D-Set rear park aid chime volume level while no other chime is active

<b>Actors</b>	Vehicle Occupant
<b>Pre-conditions</b>	Infotainment System is Powered ON  The HMI is configured to have Rear Park Aid chime volume adjustment HMI available when Rear Park Aid is on the vehicle.



	<p>Park Aid chime adjustment HMI is the active HMI screen</p> <p>No chime is active.</p>
<b>Scenario Description</b>	The user changes the chime volume level of the rear park aid chime via the HMI
<b>Post-conditions</b>	<p>The rear park aid sound/feedback chime is played at the new volume level.</p> <p>If infotainment audio was playing (ex AM/FM/USB...) then the audio is attenuated to the attenuation level that is normally used when playing the rear park aid chime.</p> <p>The new volume level of the rear parking aid chime is changed and stored (stored between ignition cycles).</p> <p>The new rear park aid chime volume level is shown on the HMI.</p>
<b>Notes</b>	<p>This use case was for rear park aid chime but applies to any supported adjustable chime. The attenuation signal only applies to chimes that support chime attenuation.</p> <p><a href="#">The HMI for adjustable chime settings should only be shown when ignition is in Run</a></p>
<b>Interfaces</b>	Vehicle System, G-HMI

#### 2.3.1.2 ALERT-UC-REQ-195370/D-Set front park aid chime volume level while no other chime is active

<b>Actors</b>	Vehicle Occupant
<b>Pre-conditions</b>	<p>Infotainment System is Powered ON.</p> <p>The HMI is configured to have Front Park Aid chime volume adjustment HMI available when Front Park Aid is on the vehicle</p> <p>Park Aid chime adjustment HMI is the active HMI screen.</p> <p>No chime is active.</p>
<b>Scenario Description</b>	The user changes the chime volume level of the front park aid chime via the HMI
<b>Post-conditions</b>	<p>The front park aid sound/feedback chime is played with the new volume level.</p> <p>If infotainment audio was playing (ex AM/FM/USB...) then the audio is attenuated to the attenuation level that is normally used when playing the front park aid chime.</p> <p>The new volume level of the front parking aid chime is changed and stored (stored between ignition cycles).</p> <p>The new front park aid chime volume level is shown on the HMI.</p>
<b>Notes</b>	<p>This use case was for front park aid chime but applies to any supported adjustable chime. The attenuation signal only applies to chimes that support chime attenuation.</p> <p><a href="#">The HMI for adjustable chime settings should only be shown when ignition is in Run</a></p>
<b>Interfaces</b>	Vehicle System, G-HMI

#### 2.3.1.3 ALERT-UC-REQ-195371/E-Set front/rear park aid chime volume level while another chime is playing through the same speakers (Front or Rear speakers)



<b>Actors</b>	Vehicle Occupant
<b>Pre-conditions</b>	<p>Infotainment System is Powered ON.</p> <p>The HMI is configured to have Front Park Aid chime volume adjustment HMI available when Front Park Aid is on the vehicle</p> <p>Park Aid chime adjustment HMI is the active HMI screen.</p> <p>A non front park aid chime is currently playing out of the front speakers (ex low fuel warning chime, seat belt chime...) through the infotainment system</p>
<b>Scenario Description</b>	The user changes the chime volume level of the front park aid chime via the HMI
<b>Post-conditions</b>	<p>If there is a currently active chime (ex seat belt chime) playing out of the same speakers as the front park aid chime (ie directionality equals front speakers) then the front park aid sound/feedback chime is not played.</p> <p>If there is a currently active chime (ex seat belt chime) playing out of the same speakers as the front park aid chime (ie directionality equals front speakers) then the audio attenuation associated with the front park aid chime is not sent (audio attenuation could be sent for the active chime though if that chime has audio attenuation associated with it).</p> <p>The new volume level of the front parking aid chime is changed and stored (stored between ignition cycles). This happens whether or not the front park aid feedback chime was played.</p> <p>The new front park aid chime volume level is shown on the HMI.</p>
<b>Notes</b>	<p>This use case applies to any supported adjustable chime and not just front park aid chime</p> <p><a href="#">The HMI for adjustable chime settings should only be shown when ignition is in Run</a></p>
<b>Interfaces</b>	Vehicle System, G-HMI

#### 2.3.1.4 ALERT-UC-REQ-195372/F-Set front/rear park aid chime volume level while the front/rear park aid chime is already playing

<b>Actors</b>	Vehicle Occupant
<b>Pre-conditions</b>	<p>Infotainment System is Powered ON.</p> <p>The HMI is configured to have Front Park Aid chime volume adjustment HMI available when Front Park Aid is on the vehicle.</p> <p>Park Aid chime adjustment HMI is the active HMI screen.</p> <p>The Front Park Aid chime is currently playing out of the front speakers through the infotainment system</p>
<b>Scenario Description</b>	The user changes the chime volume level of the front park aid chime via the HMI while the front park aid chime is already playing
<b>Post-conditions</b>	<p>The front park aid chime that is already playing is played with the new volume level. This does not change the chime itself other than the volume. The front park aid duration and pause between chimes is not changed with the front park aid volume adjustment.</p> <p>The new volume level of the front parking aid chime is changed and stored.</p>



	The new front park aid chime volume level is shown on the HMI.
Notes	This use case applies to any supported adjustable chime and not just front park aid chime <a href="#">The HMI for adjustable chime settings should only be shown when ignition is in Run</a>
Interfaces	Vehicle System, G-HMI

**2.3.1.5 ALERT-UC-REQ-209067/E-Adjust chime volume while adjustable chime is playing feedback chimes**

Actors	Vehicle Occupant
Pre-conditions	Infotainment System is Powered ON.  The HMI is configured to have Front Park Aid chime volume adjustment HMI available when Front Park Aid is on the vehicle  Park Aid chime adjustment HMI is the active HMI screen.  No Chimes are playing through the infotainment system
Scenario Description	1. The user changes (this case increases) the chime volume level of the front park aid chime via the HMI to +1 2. While the feedback chime is playing the user increases the chime volume level again to +2
Post-conditions	The current Chime_16 Forward Park Aid feedback chime sound completes it sound at +1 adjustable chime volume (doesn't play other Chime_16 repetitions for feedback chime at +1)  The Chime_16 Forward Park Aid feedback chime is played at the desired number of repetitions at the +2 adjustable chime volume
Notes	This use case applies to reverse park aid also and not just front park aid chime  See the Cluster Chime Arbitrator spec for this use case and in case of any conflict the chime arbitrator spec shall take precedent  <a href="#">The HMI for adjustable chime settings should only be shown when ignition is in Run</a>
Interfaces	Vehicle System, G-HMI

**2.3.1.6 ALERT-UC-REQ-233120/C-Chime event using an Adjustable Chime setting**

Actors	Vehicle Occupant
Pre-conditions	Infotainment System is Powered ON.  Front Park Aid adjustable chime volume is already set to + 1 adjustable chime value (user previously adjusted the settings in earlier ignition cycles)  No Chimes are playing through the infotainment system
Scenario Description	Front Park Aid chime event happens <ul style="list-style-type: none"><li>The Cluster request Front Park Aid chime at the +1 adjustable chime value</li></ul>
Post-conditions	Infotainment System plays the Front Park Aid chime at the +1 adjustable chime loudness level



**Notes**

This use case applies to any supported adjustable chime and not just front park aid chime

[The HMI for adjustable chime settings should only be shown when ignition is in Run](#)

**Interfaces**

Vehicle System



### 2.3.2 Requirements

For adjustable chimes the Chime ID signal requirement "Alert-SR-REQ-237862-Chime ID signal" has to be supported. Once supported in the AHU software then the Chime ID signal shall be supported by the AHU regardless whether adjustable chimes is supported or not.

#### 2.3.2.1 ALERT-SR-REQ-232946/D-Feature Based Message Protocol usage for adjustable chimes HMI Settings

The definition of how to use Feature Based Message Protocol for Set and Query operations is defined in the Feature Based Message Protocol SPSS and shall be followed to support this SPSS.

To support this SPSS for Adjustable Chimes the supplier will need an August 2016 or later version of the FBMP SPSS to accompany this spec.

~~For Adjustable Chimes the Adjustable Chime Client (ex SYNC) shall use the infotainment start-up method (ie when HMIAudioMode goes from OFF to ON) to query the adjustable chimes setting values from the Chime Client/Adjustable Chime Server (ex Cluster). This is described in the FBMP SPSS.~~

For Adjustable Chimes the Adjustable Chime Client (ex SYNC) shall use the start-up method where the Adjustable Chime Client queries the adjustable chimes setting values from the Chime Client/Adjustable Chime Server (ex Cluster) after ignition goes to Run. This is described in the FBMP SPSS.

#### 2.3.2.2 ALERT-SR-REQ-209035/E-Saving Adjustable Chime Settings between power modes

The Cluster Chime Client / Cluster Adjustable Chime Server shall remember the adjustable chime volume selection values (ex frontParkAidChimeVolSelection.St, rearParkAidChimeVolSelection.St) between power modes. This includes but is not limited to ignition cycles and between network bus wake-up and sleep cycles.

#### 2.3.2.3 ALERT-REQ-208870/E-Adjustable chimes when the Cluster is the chime audio source when chimes are not supported by the infotainment system

When chimes are not through the infotainment system and the cluster back up chime generator is the Chime Audio Source (ex AHU\_Chime\_Supported = Not\_Supported) then the chimes will be played by the Cluster at the Cluster default chime volume and will not support adjustable chime volume levels.

#### 2.3.2.4 ALERT-HMI-REQ-208871/F-HMI for adjustable chimes when chimes is not through the infotainment system

When AHU\_Chime\_Supported = Not Supported the Adjustable Chime HMI (ex APIM) shall:

- not allow the user to adjust the chime volume from the HMI adjustable chime menu.
  - HMI may want to indicate to the user that the adjustable chime feature is not supported (such as greying out the menu pick). HMI team to determine
- Show the adjustable chime volume at the 0 / default volume

The Adjustable Chime HMI Client (ex APIM) shall monitor the AHU\_Chime\_Supported signal to support this requirement.

Note: Chimes are not through the infotainment system whenever the signal AHU\_Chime\_Supported = Not Supported. When chimes are not supported by the infotainment system (ex speaker open circuit) then the Cluster back-up chime generator is the Chime Audio Source for that ignition cycle (or until the issue is fixed).

#### 2.3.2.5 ALERT-SR-REQ-208872/G-HMI when adjusting the chime volume

When the Adjustable Chime HMI Client sends a request to change the chime volume to the Cluster Chime Client / Adjustable Chime Server, the HMI shall not show the updated chime volume until the Adjustable Chime HMI Client receives the response from the cluster chime client / Adjustable Chime Server that it is at the new chime level.

Example:

Pre-Condition:

The Adjustable HMI client has displayed the Front Park Aid adjustable chime volume as 0 / default volume.

Event:

The user increases the Front Park Aid adjustable chime volume to +1 from the HMI and the Adjustable Chime HMI Client sends Feature.Rq(Operation = Set; FeatureID = FrontParkAidChimeVolSelection; Configuration = +1 Adjustable Chime Volume Level; PersonalityIndex = Active Personality) to the Cluster Chime Client / Adjustable Chime Server.

The Adjustable HMI Clients HMI still shows the Front Park Aid adjustable chime volume as 0 / default volume.

Post-Condition:

The Adjustable Chime HMI Client (ex SYNC) receives Feature.St(FeatureID = FrontParkAidChimeVolSelection; Configuration = +1 Adjustable Chime Volume Level; PersonalityIndex = Active Personality) and updates the HMI to show Front Park Aid is at +1 adjustable chime volume level.

### 2.3.2.6 ALERT-REQ-208983/E-Adjustable Chime Client HMI configuration

The Adjustable Chime Client (ex APIM) shall have a configuration for what adjustable chimes are supported or not on the HMI. When adjustable chimes are not supported the HMI for adjustable chimes shall not be shown.

### 2.3.2.7 ALERT-SR-REQ-209036/L-Mapping Adjustable Chime HMI selection to an Adjustable Chime Sound

The infotainment system chime audio source (ex AHU and DSP AMP) shall use the AdjustableChimeVol.St and AdjustableChimeVol2.St signals in the IPC\_Chime / IPC\_Chime2 message to map the adjustable chime value to their volume loudness value specified in the module hardware specification / interior harmony specification for a chime sound.

The chime audio source (ex AHU / DSP AMP) shall support adjustable chimes for the following chime sounds in the IPC\_Chime / IPC\_Chime2 message when Chime / Chime2 equals:

- Chime\_16 Forward Park Chime
- Chime\_17 Reverse Park Chime
- Chime\_27 RPA Continuous
- Chime\_28 FPA Continuous

Note: for a complete list of chime sounds see the Chime\_X ID assignment list see "Alert-REQ-014755-Chime ID Assignments" in the Chime SPSS.

Selected adjustable chime volume level on the HMI sent to the Cluster adjustable chime server	AdjustableChimeVol.St / AdjustableChimeVol2.St signals value in the IPC_Chime and IPC_Chime2 messages	Mapping loudness value for the Chime Audio Source
-3	-3	See Module Hardware Spec / interior harmony specification for loudness target
-2	-2	See Module Hardware Spec / interior harmony specification for loudness target
-1	-1	See Module Hardware Spec / interior harmony specification for loudness target
0 / default volume	0 / default volume	See Module Hardware Spec / interior harmony specification for loudness target
+1	+1	See Module Hardware Spec / interior harmony specification for loudness target
+2	+2	See Module Hardware Spec / interior harmony specification for loudness target
+3	+3	See Module Hardware Spec / interior harmony specification for loudness target



Note: the AHU and DSP AMP shall have stored the Chime\_X adjustable chime loudness values (ie -3, -2, -1, +1, +2, +3) above and below the 100% max loudness value (0 / default volume) for an adjustable Chime\_X sound.

The Chime\_Vol\_Level signal in the IPC\_Chime message shall adjust the Chime\_X volume levels for both the default level (0) and for the adjustable chime values (ie -3, -2, -1, +1, +2, +3) by the percentage indicated in the Chime\_Vol\_Level signal.

Note: for the Adjustable HMI Chime Client (ex APIM) and Adjustable Chime Server (ie Cluster) see the Feature Based Message Protocol “AdjustableChimeVolumeSelection – FBMP” signal interface descriptions in the beginning of this SPSS for what adjustable chime HMI interfaces are supported.

#### 2.3.2.8 ALERT-SR-REQ-209680/G-Setting Adjustable Chime signal in the IPC\_Chime and IPC\_Chime2 chime request messages

When the Cluster Chime Arbitrator Client sends IPC\_Chime / IPC\_Chime2 infotainment chime requests to the infotainment system to play chimes at an adjustable chime volume loudness level then the “IPC\_Chime : AdjustableChimeVol” signal value shall be set and held at the adjustable chime value for the duration of the Chime Request with the other IPC\_Chime signal parameters (ie held until No\_Chime is sent by the Chime Client).

When the Cluster Chime Client sends a No\_Chime the Cluster AdjustableChimeVol signal shall be set to “Inactive”.

- In other words the Cluster will send “IPC\_Chime (Chime = No\_Chime, AdjustableChimeVol = Inactive)”.

Note: this could be for the Cluster requesting a standard chime at an adjustable chime level or requesting a feedback chime when the user adjusts the chime volume level.

#### 2.3.2.9 ALERT-SR-REQ-209686/E-Setting Adjustable Chime signal in the AHU\_Alert and DSP\_AMP\_Alert messages

If the AHU or DSP AMP Chime Audio Source gets an IPC\_Chime / IPC\_Chime2 request to play an adjustable chime for a Chime\_X sound that the chime audio source does not support adjustable chimes for then the chime audio source shall set AdjustableChimeVolSupported = Not Supported.

When the AHU or DSP AMP chime audio source plays a chime at an adjustable chime volume loudness level the AdjustableChimeVolSupported signal shall be set for the duration of playing the requested IPC\_Chime / IPC\_Chime2 chime.

When a DSP AMP is present the AHU shall monitor the DSP\_AMP\_Alert message and when the DSP AMP responds with “Alert\_Chane = Initialized for chimes” the AHU shall:

- set “AHU\_Alert : AdjustableChimeVolSupported = Supported” if the DSP AMP sets “AdjustableChimeVolSupported = Supported”, else
- set “AHU\_Alert : AdjustableChimeVolSupported = Not Supported” if the DSP AMP sets “AdjustableChimeVolSupported = Not Supported”
- continue to keep “AHU\_Alert : AdjustableChimeVolSupported = Inactive” if the DSP AMP incorrectly keeps set “AdjustableChimeVolSupported = Inactive”

When the adjustable chime is no longer being played the AdjustableChimeVolSupported signal shall be set back to Inactive

- AHU: (AHU\_Alert : AHU\_Chime\_Active = No\_Chime; AdjustableChimeVolSupported = Inactive).
- DSP AMP: (DSP\_AMP\_Alert : Alert\_Channel = Inactive; AdjustableChimeVolSupported = Inactive)

The Cluster only monitors the AHU AdjustableChimeVolSupported signal and does not look at the DSP AMP signal.

See sequence diagrams for examples

#### 2.3.2.10 ALERT-SR-REQ-209764/G-Error Handling when infotainment system doesn't support adjustable chimes for a particular chime sound

If the Chime/Chime2 chime\_x sound signals in the IPC\_Chime/IPC\_Chime2 messages has the AdjustableChimeVol signal set to a value other than “0 adjustable chime volume / default volume” for a chime\_X sound that the Chime Audio Source (AHU/DSP AMP) does not support adjustable chimes for then:



1. The Chime Audio Source shall ignore the "IPC\_Chime / IPC\_Chime2 : AdjustableChimeVol.St / AdjustableChimeVol2.St" signal for the adjustable chime sound it does not support and play the chime through the infotainment system as it normally would as indicated in the other parameters in the IPC\_Chime / IPC\_Chime2 messages (play the default volume).
  - a. The Chime Audio Source would play the chime\_x sounds as though AdjustableChimeVol = 0 adjustable chime volume / default volume.
  - b. The Chime Audio Source shall not set Chime\_Not\_Recognized which would cause the Cluster to play the chime. As long as the Infotainment Chime Audio Source supports the chime sound it shall be played by the infotainment system.
2. The Chime Audio Source (ex AHU, DSP AMP) shall set AdjustableChimeSupported or AdjustableChimeSupported2 = Not Supported for the chime sound (ex Chime\_X in "Alert-014755\_Chime ID Assignments" table) that it does not support adjustable chimes for while it is playing the requested chime.
3. If the DSP AMP is the Chime Audio Source and if the DSP AMP AdjustableChimeVol / AdjustableChimeVol2 signal equals "Not Supported" while "DSP\_AMP\_Alert : Alert\_Chan = Initialized for Chimes" then the AHU shall set:
  - a. AHU\_Alert (AHU\_Chime\_Active / AHU\_Chime\_Active2 = Chime\_X) for the chime it is playing through the infotainment system for the chime sound the DSP AMP doesn't support, and
  - b. At the same time set AHU\_Alert (AdjustableChimeVolSupported / AdjustableChimeVolSupported2) to Not\_Supported.
4. When the Cluster Chime Client receives AHU\_Alert : AdjustableChimeSupported / AdjustableChimeSupported2 = Not\_Supported then the Cluster Chime Client shall:
  - a. set 0 / Default Chime in the Feature.St message so the HMI is updated to show that the chime volume is not staying at the adjustable chime volume level selected by the user but at the default chime volume.
    - i. Note: Cluster would have to update the FBMP Feature.St(FeatureID = Applicable Adjustable Chime; Configuration = 0 / default adjustable chime volume) to update the Adjustable Chime Client HMI to show that still at the default chime volume.

Note: See sequence diagram

"Alert-SD-209691-User tries to change adjustable chime value for a chime sound the infotainment system doesn't support adjustable chimes for" for details of the order the signals are sent.



## 2.3.3 Sequence Diagrams

### 2.3.3.1 ALERT-SD-REQ-209690/G-User sets an adjustable chime loudness value for a chime

#### Pre-condition:

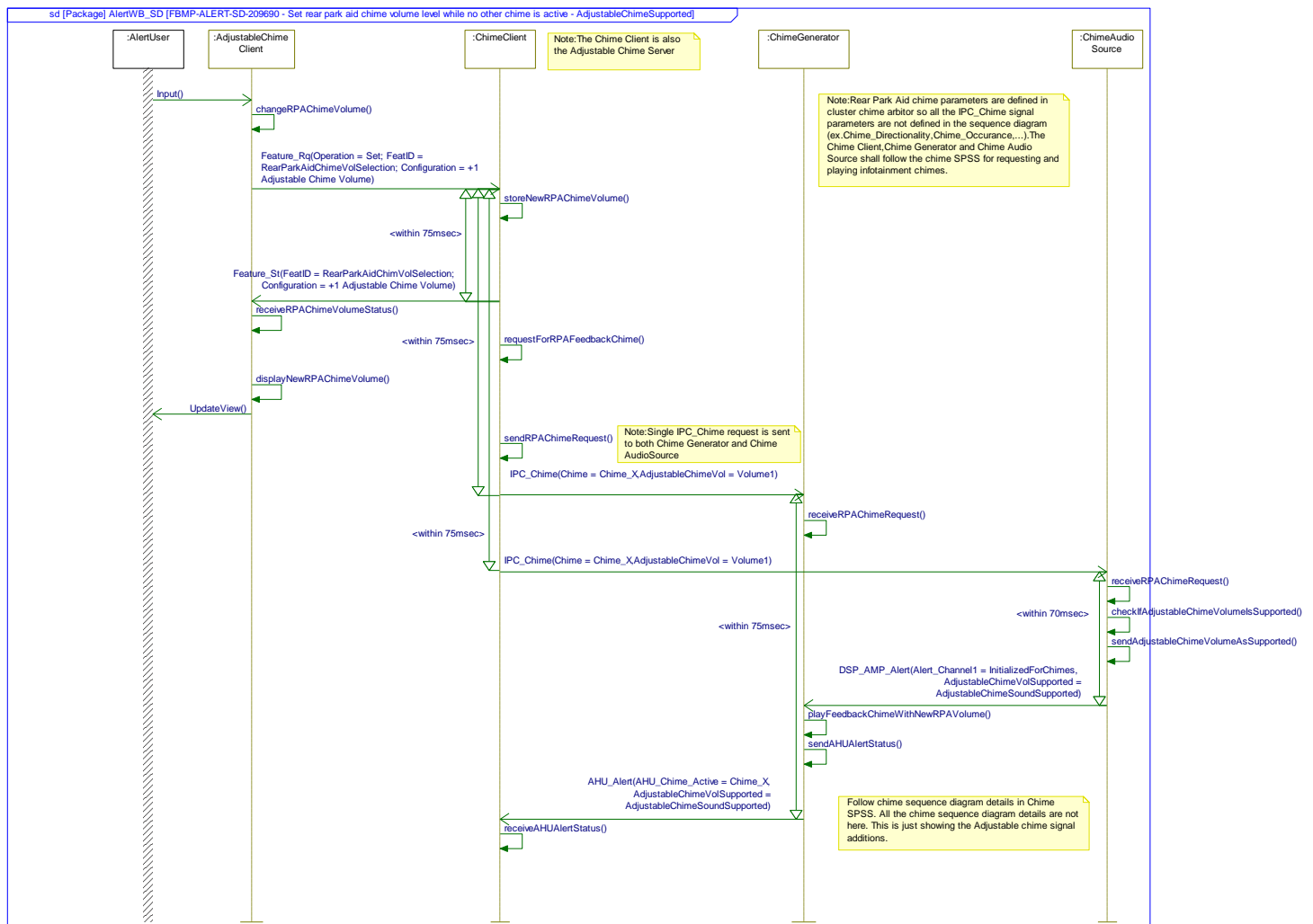
Adjustable chimes HMI screen is shown  
Adjustable chimes is set to 0 / default volume loudness value for Reverse Park Aid (note: says RPA as an example but applies for any adjustable chime)  
No chime is active

#### Event:

The user changes the reverse park aid chime loudness value from 0 / default to +1

#### Post-Condition:

The RPA feedback Chime is played at the +1 adjustable chime value  
The HMI shows RPA chime at the +1 adjustable chime value





### 2.3.3.2 ALERT-SD-REQ-209691/F-User tries to change adjustable chime value for a chime sound the infotainment system doesn't support

**Pre-condition:**

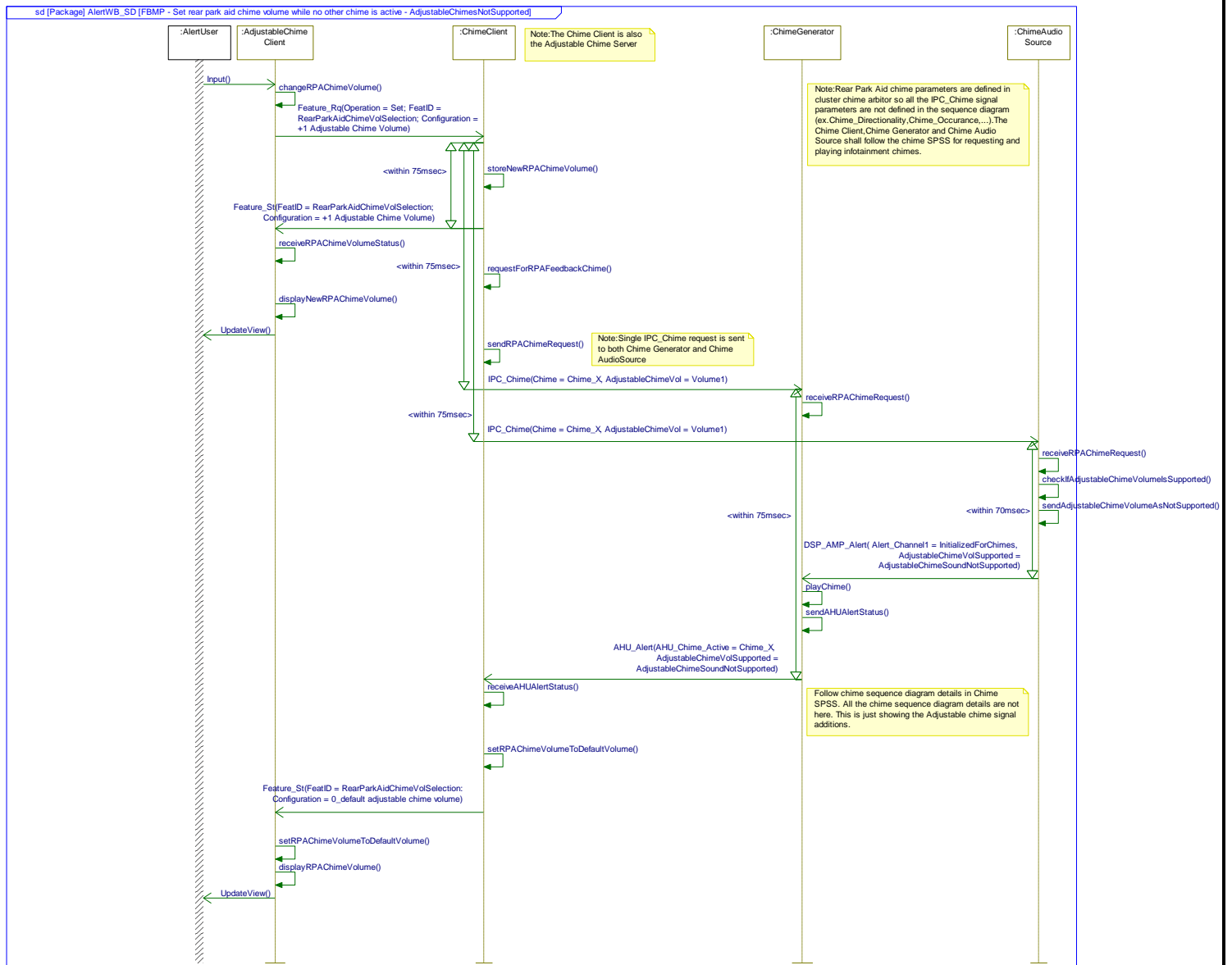
Adjustable Chimes HMI screen is shown  
Adjustable chimes is set to 0 / default volume loudness value for Reverse Park Aid (note: says RPA as an example but applies for any adjustable chime).  
No Chime is active

**Event:**

The user changes the reverse park aid chime loudness value from 0 / default to +1

**Post-Condition:**

RPA feedback chime is played at the default volume  
HMI is shown at the RPA default volume





**2.3.3.3 ALERT-SD-REQ-233176/B-Normal Chime Event using the adjustable chime signal****Pre-condition:**

No Chime is active

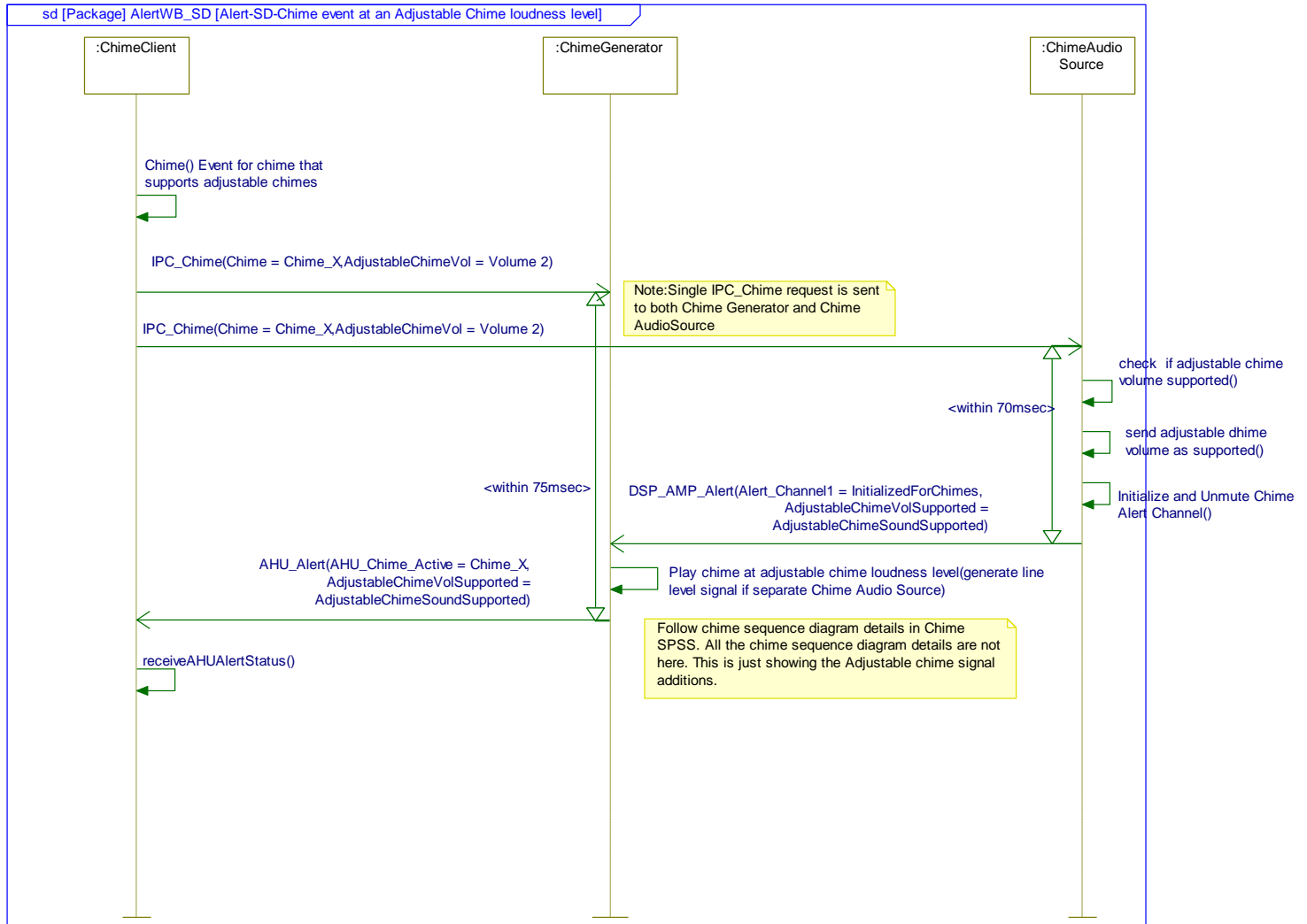
On a previous ignition cycle adjustable Chime\_X was set to +2 adjustable chime volume

**Event:**

A chime event occurs on the vehicle in which Chime\_X is sent from the Cluster to the infotainment system with the Chime\_X adjustable chime volume set to +2

**Post-Condition:**

The chime is played by the infotainment system at the +2 level indicated in the Adjustable Chime signal from the Cluster





## 2.4 ALERTv3-FUN-REQ-414038/A-Mixable Prompts - Variant 3

### 2.4.1 Overview - mixable prompts

For mixable prompts reference audio management function "[AUMGNTv2-FUN-410867-Mixable Prompts](#)".

Many mixable prompts have audio attenuation of other non-prompt audio while the mixable prompt(s) are playing. The audio attenuation is handled internal to the PDC and the volume status signal will reflect the attenuated audio. Reference the volume SPSS audio attenuation requirements below:

- "[VOLv-REQ-412097-Volume Attenuation / Restoration – variant2](#)"
- "[VOL-REQ-412869-Volume Audio Attenuation Event](#)"
- "[VOLv2-REQ-412872-Audio Attenuation event ending](#)"

For the logical internal audio attenuation strategy for Audio Attenuation please reference "[ALERT-FUN-014794-Audio Attenuation/Muting Strategy](#)".



## 2.5 ALERTv2-FUN-REQ-414037/A-Beeps

### 2.5.1 Overview - Beeps

On the Phoenix architecture, APIM PDC beeps are internally generated by the APIM PDC (Phoenix Domain Controller) with no CAN signal interaction.

Reference audio routing for beeps reference "[VOLv2-REQ-412367-Module Deployment and Audio Routing](#)".

Note: for specific features if modules external to the APIM PDC request beeps, then reference those specifications for details on the CAN interaction.



## 2.6 ALERT-FUN-REQ-414530/A-Audio Attenuation Phoenix Architecture - Captains Announcement and ICC

### 2.6.1 Requirements

#### 2.6.1.1 ALERT-SR-REQ-414521/A-Audio Attenuation - APIM PDC

The APIM PDC shall set the audio attenuation signals Attn\_Info\_Audio and New\_Attn\_Event in the SYNC\_Alerts and IPC\_Infotainment messages for any audio attenuation event including Chime and Mixable Prompt attenuation. The APIM PDC shall follow audio attenuation requirements in ALERT-FUN-014794-Audio Attenuation / Muting Strategy.

- The VMCU will control the IPC\_Infotainment chime audio attenuation
- The CCPU will control the SYNC\_Alerts mixable prompt audio attenuation

For Chime and Mixable Prompt attenuation the APIM PDC will also set attenuated volume levels in the volume status signals (ex Audio\_Vol\_Level) for any audio attenuation event including Chimes and Mixable Prompts. Reference the volume audio attenuation requirements in the Volume SPSS:

- "VOLv2-REQ-412097-Volume Attenuation / Restoration – variant 2".
- "VOL-REQ-412869-Volume Audio Attenuation Event"
- "VOLv2-REQ-412872-Audio Attenuation event ending"

#### 2.6.1.2 ALERT-SR-REQ-414529/A-Audio Attenuation Captains Announcement and ICC - PAC and DSP AMP

The PAC / DSP AMP shall support the attenuation signals Attn\_Info\_Audio and New\_Attn\_Event in the SYNC\_Alerts and IPC\_Infotainment messages ONLY for Captains Announcement and In Car Communication when Captains Announcement and In Car Communication is controlled by the PAC module. The only audio the PAC/DSP AMP would attenuate or mute would be the Captains Announcement and ICC audio from the PDC.

- Reference requirements:
  - "VOLv2-REQ-412097-Volume Attenuation / Restoration – variant 2"
  - ALERT-FUN-014794-Audio Attenuation / Muting Strategy

The PAC / DSP AMP shall NOT support audio attenuation from the Attn\_Info\_Audio and New\_Attn\_Event audio attenuation signals for any feature other than PAC generated Captains Announcement and In Car Communication. This includes not supporting Attn\_Info\_Audio and New\_Attn\_Event audio attenuation signals for volume (ie Media, Phone, Call Ring, RA and Prompt). For volume attenuation reference the requirements in the Volume SPSS:

- "VOLv2-REQ-412097-Volume Attenuation / Restoration – variant 2".
- "VOL-REQ-412869-Volume Audio Attenuation Event"
- "VOLv2-REQ-412872-Audio Attenuation event ending"



## 2.7 ALERT-FUN-REQ-014794/D-Audio Attenuation/Muting Strategy (TcSE ROIN-119789-1)

There may be multiple audio attenuation / muting requests from outside of the Infotainment System for chimes (ex. FCW, RPA...) but the [AHU / DSP AMP Volume Setting Server](#) shall only support 1 audio attenuation / muting request signal from outside the Infotainment System for chimes. The 'IPC\_Infotainment.St() : Attn\_Info\_Audio' signal shall be used to Attenuate / Mute the infotainment audio for chimes. The Cluster is the Audio Attenuation Request Prioritizer responsible for sending the audio attenuation requests from outside the Infotainment System to the [AHU / DSP AMP Volume Settings Server](#) for chimes.

- Note: Regardless of whether the Chime\_Source is the Cluster or Infotainment System the Audio Attenuation Requests are independent of chimes and chime source and shall always be supported.

There may also be muting requests internal to the infotainment system such as for mixable prompts using the 'SYNC\_Alerts : Attn\_Info\_Audio' signal.

For the Phoenix architecture with the Cluster, APIM, Chime Generator, Chime Audio Source integrated in the APIM PDC then internally it could be considered that logically the VMCU (cluster chime software) sends the CCPU the audio attenuation signals as described in this function/section so the volume signals sent by APIM PDC could be at attenuated levels. In this case the CCPU would be treated as the Audio Attenuation Server/Volume Settings Server as described in this function/section for audio attenuation purposes.

### 2.7.1 Cluster Prioritizing Audio Attenuation Requests

#### 2.7.1.1 ALERT-SR-REQ-014795/C-Cluster Prioritizing Audio Attenuation Requests (TcSE ROIN-39883-2)

The Cluster may receive multiple audio attenuation requests at the same time from outside the infotainment system. While multiple audio attenuation requests are active at the same time the Cluster shall use the request with the greatest attenuation of audio for the Attn\_Info\_Audio signal. The Cluster Attn\_Info\_Audio signal shall be broadcast on the Infotainment bus.

For example, while FCW (forward collision warning) is active with a full audio mute if a Reverse Park Aid signal is requesting the audio be partially attenuated then the audio will remain fully muted since FCW has the highest attenuation level.

Upon exiting the audio attenuation / muting request with the greatest attenuation, the next highest audio attenuation request shall be serviced until there are no longer any active audio attenuation requests.

### 2.7.2 AHU / DSP AMP Prioritizing Audio Attenuation Requests

#### 2.7.2.1 ALERT-SR-REQ-014796/D-Multiple Attenuation Requests (TcSE ROIN-39884-2)

The [AHU / DSP AMP Volume Settings Server](#) may receive multiple audio attenuation requests at the same time whether from outside the infotainment system (ex. chime requests sent to the Cluster which sends attenuation requests to infotainment components) or internal to the infotainment system (ex. SYNC prompts attenuation requests). While multiple audio attenuation requests are active at the same time the [AHU / DSP AMP Volume Settings Server](#) shall use the request with the greatest attenuation of audio.

Upon exiting the audio attenuation / muting request with the greatest attenuation, the next highest audio attenuation request shall be serviced until there are no longer any active audio attenuation requests.

### 2.7.3 Infotainment System Audio Attenuation

#### 2.7.3.1 ALERT-SR-REQ-014797/D-Audio Attenuation to Attenuation\_X volume level (TcSE ROIN-39889-2)

When the [AHU / DSP AMP Volume Settings Server](#) receives the signal Attn\_Info\_Audio with New\_Attn\_Event = Active then the [AHU / DSP AMP Volume Settings Server](#) will lower their infotainment volume to Attenuation\_X audio level as defined in the module component requirements (see requirement "VOL-REQ-014819-Volume Attenuation/Restoration"). For the Phoenix architecture the see requirement "VOLv2-REQ-412097-Volume Attenuation / Restoration – variant 2".

The Transmitter Audio Attenuation Client shall set Attn\_Info\_Audio to the desired attenuation level for the duration of the requested audio attenuation. See "ALERT-REQ-014804-Transmitters usage of New\_Attn\_Event signal" for the usage of the New\_Attn\_Event signal.

Ex.



1. Audio Attenuation Client sends an attenuation request with Attn\_Info\_Audio = Attenuation\_X / New\_Attn\_Event = Active
2. Audio Attenuation Client sends Attn\_Info\_Audio = Attenuation\_X / New\_Attn\_Event = Inactive (See requirement "[ALERT-REQ-014804-Transmitters usage of New\\_Attn\\_Event](#)" for details).
3. Audio Attenuation Client keeps sending Attn\_Info\_Audio = Attenuation\_X / New\_Attn\_Event = Inactive while the attenuation event is occurring (note: Audio Attenuation Client has no knowledge if a partial attenuation is cancelled).
4. When Audio Attenuation Client no longer needs the attenuation it shall send Attn\_Info\_Audio = 'No Attenuation of Audio' / New\_Attn\_Event = Inactive.

#### 2.7.3.2 [ALERT-SR-REQ-014798/D-Audio Attenuation Timing \(TcSE ROIN-39890-2\)](#)

The [AHU/DSP AMP Volume Setting Server](#) shall lower their volume to the predefined audio level within 100 msec of receiving the Attn\_Info\_Audio signal with the New\_Attn\_Event = Active if the Attn\_Info\_Audio signal causes an attenuation event.

The [AHU/DSP AMP Volume Setting Server](#) shall lower their volume assuming that the present audio volume level is higher than the Attenuation\_X level, otherwise the audio volume and XXX\_Volume\_Level signal shall not change.

#### 2.7.3.3 [VOL-SR-REQ-014799/D-Volume signal update for Audio Attenuation \(TcSE ROIN-39891-2\)](#)

The [AHU/DSP AMP Volume Setting Server](#) shall update the XXX\_Volume\_Level signal after receiving the Attn\_Info\_Audio signal that results in a decrease in audio.

Note: the corresponding XXX\_Volume\_Updated signal with the XXX\_Volume\_Level signal shall be set to "No Update" for an Attn\_Info\_Audio attenuation event that changes the XXX\_Volume\_Level signal.

#### 2.7.3.4 [ALERT-SR-REQ-014800/D-User Volume adjustment lock-out period \(TcSE ROIN-39892-3\)](#)

During an Attn\_Info\_Audio audio attenuation event volume user adjustments won't be able to lower or increase volume until 200 msec after the [AHU/DSP AMP Volume Setting Server](#) first receives the Attn\_Info\_Audio signal set to an attenuation level with the signal New\_Attn\_Event = Active.

Only after 200 msec from first receiving Attn\_Info\_Audio = Attenuation\_X / New\_Attn\_Event = Active shall volume user adjustments cancel the partial attenuation.

#### 2.7.3.5 [ALERT-SR-REQ-014801/F-Volume user adjustments during a partial attenuation \(TcSE ROIN-39893-4\)](#)

Volume user adjustments are allowed after 200 msec from first receiving the Attn\_Info\_Audio signal with the signal New\_Attn\_Event = Active. Volume user adjustments to any of the attenuated volume sources shall effectively cancel the active partial attenuation attenuating the affected volume sources.

Ex.

1. Pre-Condition: XXX\_Volume\_Level = 20
2. Audio Attenuation Clients sends a partial Attenuation request with Attenuation\_X / New\_Attn\_Event = Active which sets the volume level to volume step 5
3. Then the Audio Attenuation Client sets Attn\_Info\_Audio = Attenuation\_X / New\_Attn\_Event = Inactive
4. The Audio Attenuation Server volume goes to XXX\_Volume\_Level = 5 and XXX\_Volume\_Updated = No Update.
5. The user adjust the attenuated volume up with SetVolume = +1 step while the partial attenuation is active
6. The partial attenuation is then cancelled and volume is increased to volume step 6 (XXX\_Volume\_Level = 6)

Note: If two partial attenuations are active at once then cancelling the partial attenuation of one partial attenuation doesn't necessarily cancel the attenuation of another partial attenuation.

Ex.

Pre-Condition:

1. There is both a Chime (IPC\_Infotainment) AND Prompt partial attenuation event active at the same time.
2. A media source (ex AM/FM) is granted in the Resource Update message ([AudioSource.St for Phoenix architecture](#)) and Prompts are active. Since prompts are active they are the active volume source in this case.

Event:



User adjust the volume while prompts are active

Post-Condition:

The chimes partial attenuation (IPC\_Infotainment : Attn\_Info\_Audio) is cancelled so prompts are no longer attenuated but the media source still is attenuated from the Prompt partial attenuation (SYNC\_Alerts : Attn\_Info\_Audio). The prompt partial attenuation is active so media is still attenuated.

**2.7.3.6 ALERT-SR-REQ-014802/D-Volume user adjustments during a full mute (TcSE ROIN-39894-2)**

During an Attn\_Info\_Audio attenuation event volume user adjustments shall be ignored while a full mute Attenuation event is active unless otherwise noted in the component requirements. The [AHU/DSP AMP Volume Setting Server](#) shall update the XXX\_Volume\_Level signal accordingly.

**2.7.3.7 ALERT-SR-REQ-014803/D-New Attenuation request at the same attenuation level that was cancelled because of a volume user adjustment (TcSE ROIN-39895-2)**

To protect for the case were another Attn\_Info\_Audio attenuation event happens requesting a partial attenuation at the same Attn\_Info\_Audio attenuation level that the [AHU/DSP AMP Volume Settings Server](#) is ignoring (because of a previous user volume adjustment) then to make sure the attenuation is still acted on the [AHU/DSP AMP Volume Settings Server](#) shall monitor the "New\_Attn\_Event".

If the [AHU/DSP AMP Volume Settings Server](#) receive the "New\_Attn\_Event = Active" concurrently with "Attn\_Info\_Audio" signal set to an attenuation level then the audio shall be attenuated (if applicable per other attenuation requirements) even if "Attn\_Info\_Audio" signal attenuation level did not change.

**2.7.3.8 ALERT-SR-REQ-014804/D-Transmitters usage of New\_Attn\_Event signal (TcSE ROIN-39896-3)**

The Audio Attenuation Transmitting module shall set the "New\_Attn\_Event = Active" when it requests a new Attenuation event.

The "New\_Attn\_Event" signal shall be set equal to "Active" and then set to "Inactive" (within 50 msec) and then all subsequent periodic status updates shall set the signal "New\_Attn\_Event = Inactive" until a new Attenuation event occurs.

**Ex. Audio Attenuation Client requesting the same audio attenuation at different times:**

1. The Audio Attenuation Client (audio attenuation transmitting module) requests a new attenuation event when a Reverse Park Aid chime begins. The Audio Attenuation Client sends a partial attenuation request with Attn\_Info\_Audio = Attenuation\_X and New\_Attn\_Event = Active.
2. Then the Audio Attenuation Client keeps Attn\_Info\_Audio equal to Attenuation\_X and sets New\_Attn\_Event = Inactive
3. the Audio Attenuation Client keeps Attn\_Info\_Audio equal to Attenuation\_X for as long as the Reverse Park Aid beeps are playing but keeps New\_Attn\_Event = Inactive
4. The Audio Attenuation Client requests the Seat Belt chime to play out of the front speakers while the Reverse Park Aid chime continues to play out of the rear speakers. The Audio Attenuation Clients sends a partial attenuation request with Attn\_Info\_Audio = Attenuation\_X and New\_Attn\_Event = Active.
5. Then the Audio Attenuation Client keeps Attn\_Info\_Audio equal to Attenuation\_X and sets New\_Attn\_Event = Inactive
6. the Audio Attenuation Client keeps Attn\_Info\_Audio equal to Attenuation\_X for as long as the Seat Belt chime or Reverse Park Aid chime are playing but keeps New\_Attn\_Event = Inactive
7. After some time the Reverse Park Aid chime ends and the Audio Attenuation Client keeps Attn\_Info\_Audio = Attenuation\_X and keeps New\_Attn\_Event = Inactive.
8. Some time after the Reverse Park Aid chime ends the Seat Belt chime ends and the Audio Attenuation Client sets Attn\_Info\_Audio = 'No Attenuation of Audio' and keeps New\_Attn\_Event = Inactive.

Note: The seat belt chime and RPA chime are used as examples and may or may not be the same attenuation level. Please reference the applicable function specifications for what attenuation to use





### 2.7.3.9 ALERT-SR-REQ-014805/H-Volume when changing to a lower attenuation / no attenuation from a higher attenuation (TcSE ROIN-39897-2)

When the 'Attn\_Info\_Audio = Attenuation\_X' signals change to a lower attenuation level or to no attenuation resulting in an increase in volume then the [AHU / DSP AMP Volume Settings Server](#) shall gracefully increase the volume at a rate defined in Volume SPSS requirement "[VOL-REQ-088208-Audio Attenuation / Mute Ramps](#)". [For the Phoenix architecture reference Volume SPSS requirement "VOLv3-REQ-412277-Audio Attenuation Mute / Unmute Ramps – variant 3"](#).

Upon exiting the audio attenuation / muting request with the greatest attenuation, the next highest audio attenuation request shall be serviced until there are no longer any active audio attenuation requests.

Example from two different modules sending different attenuation values in different network messages:

1. Pre-Condition: XXX\_Volume\_Level = 20
2. Audio Attenuation Client A sends a partial Attenuation request with Attenuation\_X / New\_Attn\_Event = Active which sets the volume level to volume step 16 (see attenuation table for actual volume steps to use – ie "[VOL-REQ-014819-Volume Attenuation/Restoration](#)", [for Phoenix "VOLv2-REQ-412097-Volume Attenuation / Restoration"](#))
3. Then the Audio Attenuation Client A sets Attn\_Info\_Audio (Module A) = Attenuation\_X / New\_Attn\_Event = Inactive
4. The Audio Attenuation Server volume goes to XXX\_Volume\_Level = 16 and XXX\_Volume\_Updated = No Update.
5. Audio Attenuation Client B sends a partial Attenuation request with Attenuation\_Y / New\_Attn\_Event = Active which sets the volume level to volume step 7
6. Then the Audio Attenuation Client B sets Attn\_Info\_Audio (Module B) = Attenuation\_Y / New\_Attn\_Event = Inactive
7. The Audio Attenuation Server volume goes to XXX\_Volume\_Level = 7 and XXX\_Volume\_Updated = No Update.
8. Then the Audio Attenuation Client B sets Attn\_Info\_Audio (Module B) = No Attenuation of Audio / New\_Attn\_Event = Inactive
9. The Audio Attenuation Server volume goes to XXX\_Volume\_Level = 16 and XXX\_Volume\_Updated = No Update.
10. Then the Audio Attenuation Client A sets Attn\_Info\_Audio (Module A) = No Attenuation of Audio / New\_Attn\_Event = Inactive
11. The Audio Attenuation Server volume goes to XXX\_Volume\_Level = 20 and XXX\_Volume\_Updated = No Update.

### 2.7.3.10 ALERT-SR-REQ-014806/D-Timing to increase volume because of a lowering of the attenuation level (TcSE ROIN-39898-2)

When the 'Attn\_Info\_Audio = Attenuation\_X' signals change to a lower attenuation level or to no attenuation resulting in an increase in volume then the [ACM / DSP AMP Volume Settings Server](#) shall start to increase the volume level no later than 200 msec after receiving the Attenuation update.



### 3 Appendix: Reference Documents

Reference #	Document Title
1	Chime sound characteristics reference Hardware spec
2	Reference IDS for chime diagnostics
3	Reference SYNC internal specs regarding what mixable prompts to play and when. Reference A22b and A22c
4	When Cluster is integrated with SYNC then reference the Cluster chime arbitrator specification for additional Cluster functionality
5	HMI specs if adjustable chimes are supported for adjustable chime HMI