Low Energy Audio – Context Types and Availability

***Bluetooth*® Informational Publication**

- **Version:** v1
- **Version Date:** 2023-08-29
- **Prepared By: Generic Audio Working Group**

If you have any questions or comments on this informational publication or Bluetooth LE Audio specifications, please submit a ticket: https://www.bluetooth.com/leaudio-spec-feedback/

Abstract:

This document explains how Context Type values are used to describe audio use cases and to signal availability for audio use cases.

Version History

Version Number	Date (yyyy-mm-dd)	Comments
v1	2023-08-29	Approved by the Bluetooth SIG Board of Directors.

Acknowledgments

Name	Company
Rasmus Abildgren	Bose
Georg Dickmann	Sonova AG
Nick Hunn	GN Hearing A/S
Bjarne Klemmensen	Demant A/S
Jeff Solum	Starkey Hearing Technologies
Scott Walsh	HP
Frank Yerrace	Microsoft

Contents

1 Introduction 5
1.1 Scope 5
1.2 Terminology 5
2 Context Types and availability 6
2.1 Motivation 6
2.2 Context Types as description of the purpose of an Audio Stream 6
2.3 Association of Audio Streams with Context Type values 6
2.4 Signaling of support and availability for Context Type values 7
2.4.1 The special significance of the <<Unspecified>> Context Type value 9
2.5 Signaling of preferences for audio configurations 9
2.6 Summary on data structures that hold Context Type values 10
3 References 12

1 Introduction

This document helps developers understand how and when Context Type values should be used. While each of the Low Energy (LE) Audio specifications addresses only individual aspects and rules on the use of Context Type values, this document provides an overview of their usage from the different LE Audio specifications in a single document.

This is not a Bluetooth specification, therefore, the established Bluetooth SIG specification language conventions for use of the words ***shall***, ***shall not***, ***must***, ***should***, ***should not***, ***may***, and ***can*** do not apply to this document.

1.1 Scope

This document explains the concept behind Context Type values and the signaling of the availability of devices for audio streaming, which should be expected to change during normal operation. While the document points to various LE Audio specifications, it does not add, change, or replace any mandatory rules defined by any referenced specification.

1.2 Terminology

Data structures holding Context Type values are defined in the Basic Audio Profile (BAP) Specification [6] and the Published Audio Capabilities Service (PACS) Specification [5]. The Common Audio Profile (CAP) Specification [2] adds rules on the use of Context Types on top of those already defined in BAP. This document consistently refers to the roles of (CAP) Initiator (see Section 2.1.2 in CAP [2]) and (CAP) Acceptor (see Section 2.1.1 in CAP [2]), including places where the roles may be substituted by the more general roles of BAP Unicast Client and BAP Unicast Server.

2 Context Types and availability

2.1 Motivation

The commonly used Bluetooth Classic Audio specifications, i.e., the Hands Free Profile (HFP) Specification and the Advanced Audio Distribution Profile (A2DP) Specification, have no mechanism to explicitly associate an Audio Stream with the purpose it is serving. Implementations have therefore resorted to heuristics.

For example, when an AT-command, as defined in HFP, signals a call state of 'Incoming' while an Extended Synchronous Connection Oriented (eSCO) link is established, a peripheral can conclude that the audio transported over the eSCO link is a ringtone. Similarly, audio transported over the Audio Video Distribution Transport Protocol (AVDTP) Specification is expected to be media audio.

Some implementations may conclude from the presence or absence of the Audio Video Remote Control Profile (AVRCP) Specification, signaling in conjunction with audio being transported over AVDTP, that the audio being streamed is a system-generated alert or a user-initiated media stream. Such heuristics are used by peripheral implementations to decide whether to accept or reject the establishment of a new Audio Stream from a second central device in situations where the Peripheral was already engaged in an Audio Stream with a first Central device. The lack of any specified method to link an Audio Stream with its purpose is part of the multi-profile issue inherent in Classic Audio.

The LE Audio specifications introduced the concept of Context Types to replace such heuristics by explicit signaling.

2.2 Context Types as description of the purpose of an Audio Stream

Context Type values (as defined in the Bluetooth Assigned Numbers [1]) describe the current or proposed use case (that is, the purpose) of an Audio Stream. For example, an Initiator associates the Context Type value <<Ringtone>> with an Audio Stream that alerts the listening user to an incoming voice or video call.

The Context Type values assigned to an Audio Stream are independent of the content of the Audio Stream. For example, an Audio Stream assigned with the Context Type value of <<Ringtone>> can contain a person's favorite song, a voice announcing the caller's name, or a ringing bell. The Context Type value is an indication of the purpose of the Audio Stream.

An Audio Stream can serve multiple use cases at the same time. Therefore, an Audio Stream can be associated with multiple Context Type values. To indicate that an Audio Stream is being used for more than one purpose, each Context Type value is represented by a single bit within a bitfield. For example, a phone may send an Audio Stream, which is an audible mix of audio from a media player and a ringtone signaling an incoming call. Therefore, the phone sets the Context Type values of <<Media>> and <<Ringtone>> in the bitfield for the duration of time that the streaming media includes the ringtone for an incoming call. When the ringtone stops, the phone resets the bit for <<Ringtone>> to zero.

2.3 Association of Audio Streams with Context Type values

An Initiator sets the bitfield of Context Type values in the Streaming_Audio_Contexts Metadata Length-Type-Value (LTV) structure when configuring a broadcast Audio Stream, when enabling an Audio Stream Endpoint (ASE) for a unicast Audio Stream, or when updating the metadata of a broadcast or unicast Audio Stream.

When the purpose of an Audio Stream changes, the Initiator updates the Context Type values in the Streaming_Audio_Contexts Metadata LTV structure to reflect the modified use case of the Audio Stream. The Acceptor can use this information to adapt its behavior to the use case. For example:

- An Acceptor can set its availability (see Section 2.4) as a function of the Context Type values associated with an Audio Stream that it currently maintains. For example, while maintaining an Audio Stream with a first Initiator for the <<Conversational>> use case, as in a phone call, an Acceptor can be unavailable for the <<Media>> use case, avoiding the audio of an ongoing phone call from being interrupted by a media audio stream from another Initiator.
  Such preferential handling of use cases is particularly useful when an Acceptor is not able to simultaneously maintain Audio Streams with multiple Initiators (for example, by disconnecting an established Audio Stream when accepting a new Audio Stream from another Initiator).
- An Acceptor can tailor its audio processing to a use case. For example, an Acceptor could automatically enable active noise cancelation while receiving an Audio Stream for the <<Emergency alarm>> use case such that the user can clearly hear an alarm in a noisy environment.

Sections 7.1.2.1 and 7.1.2.2 in CAP [2] provide guidance on which values for Streaming_Audio_Contexts Metadata LTV structure that an Initiator will set when audio content is controlled by instances of the Media Control Service (MCS) Specification [3], Generic Media Control Service (GMCS) Specification [3], Telephone Bearer Service (TBS) Specification [4], or Generic Telephone Bearer Service (GTBS) Specification [4].

2.4 Signaling of support and availability for Context Type values

An Acceptor signals availability for Audio Streams associated with specific Context Type values to Initiators. Acceptors separately signal availability for reception and transmission of Audio Streams. For example, a headset independently signals its availability for transmission of audio picked up by its boom microphone and for reception of audio to be rendered on its ear cups. Sections 5.6.3 and 5.6.4 in BAP [6] does not allow Initiators to enable or update a unicast Audio Stream for a Context Type value for which an Acceptor is not available. For example:

- An Acceptor that is currently receiving an Audio Stream for the <<Conversational>> use case and does not want to be interrupted by the ringtone from a call on a different phone can make itself unavailable for the <<Ringtone>> use case. Section 7.1.2.3 in CAP [2], describes how an Acceptor can resort to generation of an out-of-band ringtone in this situation.
- A hearing-impaired person wearing Bluetooth-enabled hearing aids is participating in an important meeting and does not want to be disturbed by any Audio Stream other than an <<Emergency alarm>>. The hearing aids signal availability for only <<Emergency alarm>> to any Initiator.

To unconnected Initiators, Acceptors signal availability through General or Targeted Announcements (see Section 3.5.3 in BAP [6]).

To connected Initiators, Acceptors signal availability through the Generic Attribute Profile (GATT) characteristic Available Audio Contexts (see Section 3.5 in PACS [5]).

An Acceptor can signal availability per connected Initiator through the Available Audio Contexts characteristic of PACS, exposing different values to different Initiators. For example, when a resource-constrained Acceptor that can maintain Audio Streams with only one Initiator is receiving an Audio Stream for the <<Conversational>> use case from a first Initiator, it might signal availability for all other supported Context Type values to this first Initiator, allowing it to update the use case of the already established Audio Stream. At the same time, it might signal availability for only <<Emergency alarm>> to

all other Initiators through both the PACS Available Audio Contexts characteristic and through General Announcements. This way, other Initiators will not enable an Audio Stream for any use case other than <<Emergency alarm>>, allowing uninterrupted flow of the existing <<Conversational>> Audio Stream.

An Acceptor signals support for Context Type values to Initiators to indicate for which Context Type values the Acceptor might signal availability (see Section 3.5.1 in PACS [5]). If an Initiator wants to use an Audio Stream for a use case represented by a Context Type value that an Acceptor does not support, the Initiator can use the Context Type value of <<Unspecified>> instead. The Initiator then checks the Acceptor's availability for the Context Type value of <<Unspecified>> (see Section 7.1, Table 7.1 in CAP [2]). Consequently, an Acceptor's availability for the Context Type value of <<Unspecified>> implicitly represents that Acceptor's availability for all Context Type values that the Acceptor does not support, as illustrated by Figure 2.1 and Figure 2.2.

Acceptors signal support for Context Type values through the GATT characteristic Supported Audio Contexts (see Section 3.6 in PACS [5]).

Figure 2.1 shows the example of an Acceptor that is available for everything but <<Ringtone>>, <<Notifications>>, <<Sound effects>>, <<Voice assistants>>, and <<Media>> because the Acceptor supports these Context Type values but is not available for any of them. The Acceptor is explicitly available for <<Emergency alarm>>, <<Conversational>>, and <<Unspecified>> . Because an Initiator is allowed to remap any Context Type values that the Acceptor does not support to <<Unspecified>> and the Acceptor is available for <<Unspecified>>, the Acceptor is, implicitly, also available for <<Alerts>>, <<Live>>, <<Instructional>>, <<Game>>, and any future Context Type value that has been assigned to an RFU bit.

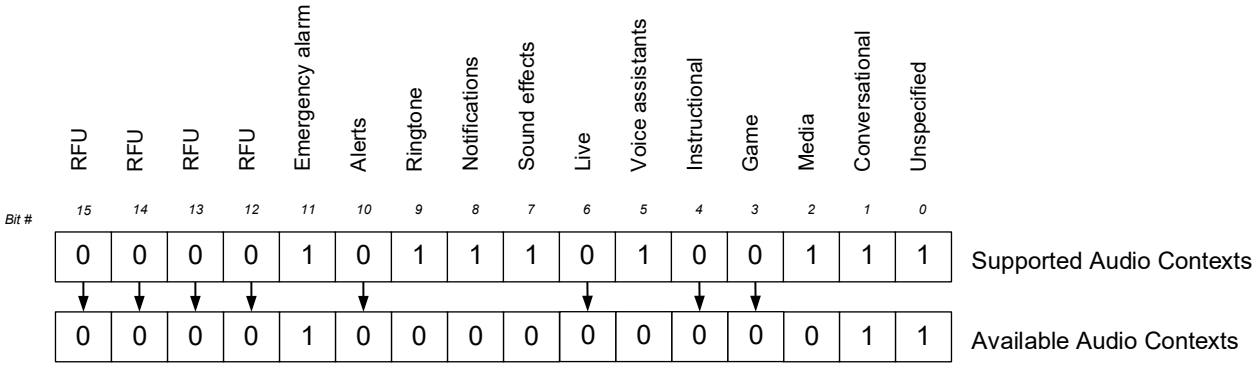

Figure 2.1: Acceptor that is available for everything but <<Ringtone>>, <<Notifications>>, <<Sound effects>>, <<Voice assistants>>, and <<Media>>

Figure 2.2 shows the example of an Acceptor that is only available for <<Emergency alarm>>, <<Ringtone>>, and <<Conversational>>.

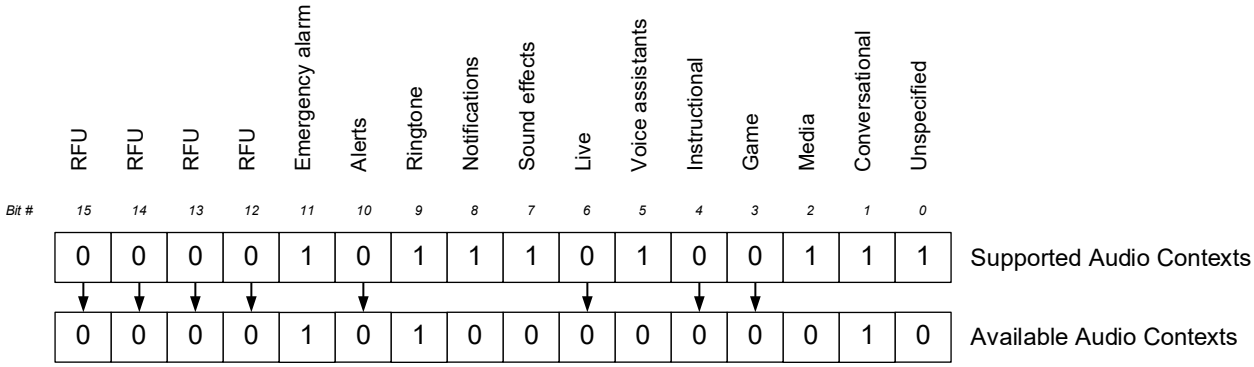

Figure 2.2: Acceptor that is only available for <<Emergency Alarm>>, <<Ringtone>>, and <<Conversational>>

Note that an Initiator is not allowed to remap a Context Type value to <<Unspecified>> when the Acceptor supports that Context Type value. It can therefore be advantageous for an Acceptor to support all Context Type values defined in [1]. For example, a dedicated call headset that does not want to engage in anything but phone calls can avoid being subjected to media, sound effects, etc. by signaling support for all Context Types while signaling availability for <<Ringtone>>, <<Conversational>>, and, possibly, <<Unspecified>> only. Because it is not allowed to set an RFU bit to 1 (see Section 1.8.2 in PACS [5]), an Initiator is not allowed to remap a future Context Type value to the value of <<Unspecified>>.

2.4.1 The special significance of the <<Unspecified>> Context Type value

The Context Type value <<Unspecified>> covers specific situations:

- Because all Acceptors support the value <<Unspecified>> (see Section 3.5.2.1 in BAP [6]), an Initiator can remap an Audio Stream with a Context Type value that an Acceptor does not support to the Context Type value <<Unspecified>> (see Section 7.1, Table 7.1 in CAP [2]). This enables Acceptors to be forward-compatible. An Acceptor that is available for the Context Type value <<Unspecified>> can accept Audio Streams from Initiators that use Context Type values that might be defined after the Acceptor was manufactured.
- Some audio devices might have encapsulated the handling of audio input and output in a software layer that was unaware of the application or use case of the audio signals. Until such architectures are adapted to become use-case aware, such Initiator implementations can associate Audio Streams with the Context Type value of <<Unspecified>>.

Note that Initiators are encouraged to use specific Context Type values rather than the value of <<Unspecified>>. Because an Acceptor that is available for a specific Context Type value may not be available for <<Unspecified>>, or vice-versa, the end user experience can be different depending on whether Initiators use specific Context Type values or <<Unspecified>>.

2.5 Signaling of preferences for audio configurations

An Acceptor can indicate preference for a Published Audio Capability (PAC) record using Context Type values. For example, a severely power-constrained device might prefer to receive media audio encoded at a low bit rate. To signal this preference, the Acceptor exposes a PAC record for the codec capability

setting 16_2 (see Section 3.5.2 in BAP [6]), adds the Preferred_Audio_Contexts Metadata LTV structure to the PAC record, and sets the Context Type value to <<Media>>. An Initiator attempts to match such preferences (See Section 7.3.1.2.4 in CAP [2]).

2.6 Summary on data structures that hold Context Type values

Table 2.1 lists the data structures that hold Context Type values.

Data Structure	Description
Streaming_Audio_Contexts Metadata LTV structure (defined in [1])	This structure describes the use case (or combination of multiple use cases) of a unicast or broadcast Audio Stream. An Initiator includes this structure in an ASE Enable operation (see Section 5.6.3 in BAP [6]), in an Update Metadata operation (see Section 5.6.4 in BAP), or in the Broadcast Audio Source Endpoint (BASE) structure (see Section 3.7.2.2 in BAP).
Preferred_Audio_Contexts Metadata LTV structure (defined in [1])	When added to a PAC record (see Section 4.3.3 in BAP [6] and Section 7.1 in CAP [2]), this structure indicates that, for the use cases described by the given Context Type values, the Acceptor prefers the configuration parameters in that PAC record.
Available Audio Contexts characteristic (defined in Section 3.5 in PACS [5])	This characteristic describes whether an Acceptor is available for a unicast Audio Stream that serves a use case described by the given Context Type values (see Section 3.5 in PACS [5]). The Available_Sink_Contexts and Available_Source_Contexts fields describe availability for each Audio Stream direction. An Acceptor can expose different values for different Initiators.
Supported Audio Contexts characteristic (defined in Section 3.6 in PACS [5])	This characteristic describes what an Acceptor supports (that is, it signals availability for the given Context Type values) (see Section 3.6 in PACS [5]). The Supported_Sink_Contexts and Supported_Source_Contexts fields describe support for each Audio Stream direction.
BAP Unicast server advertising data (AD) as General or Targeted Announcements (defined in Section 3.5.3 in BAP [6])	Advertises availability of an Acceptor to unconnected Initiators for a unicast Audio Stream serving a use case described by the given Context Type values.

Table 2.1: Data structures that hold Context Type values.

Figure 2.3 and Figure 2.4 illustrate the flow of data related to Context Type values. Devices and their roles are shown in boxes (CAP roles in bold). The flow of data is represented by arrows. The purpose of the data is labeled in bold; the data structures with Context Type information are labeled in italic.

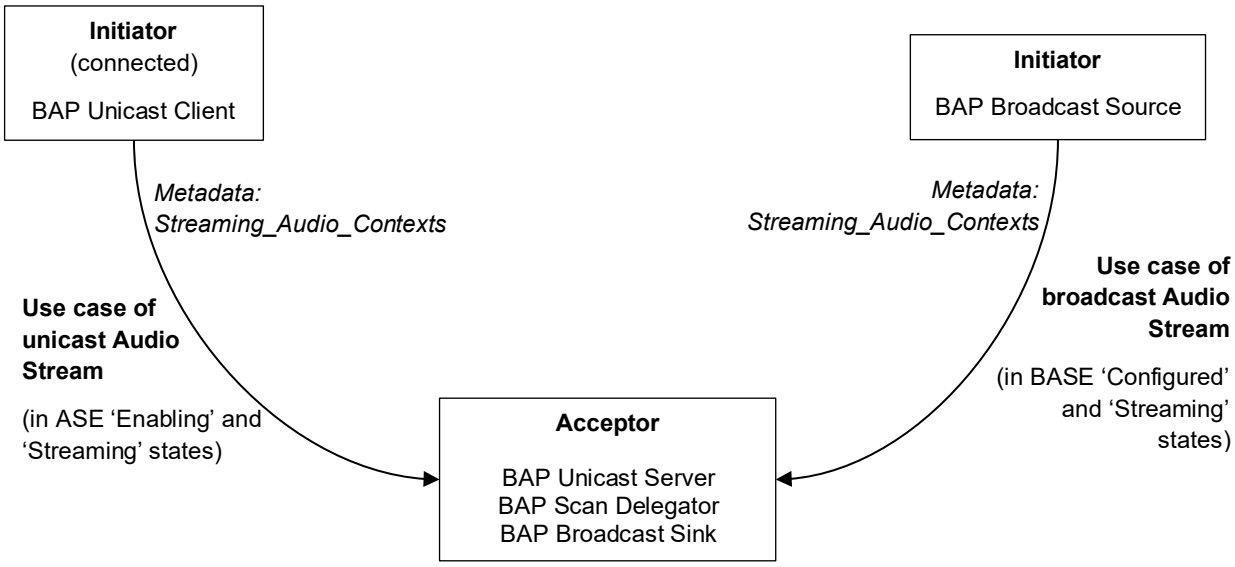

Figure 2.3: Flow of Context Type information as used to signal use cases of unicast and broadcast Audio Streams

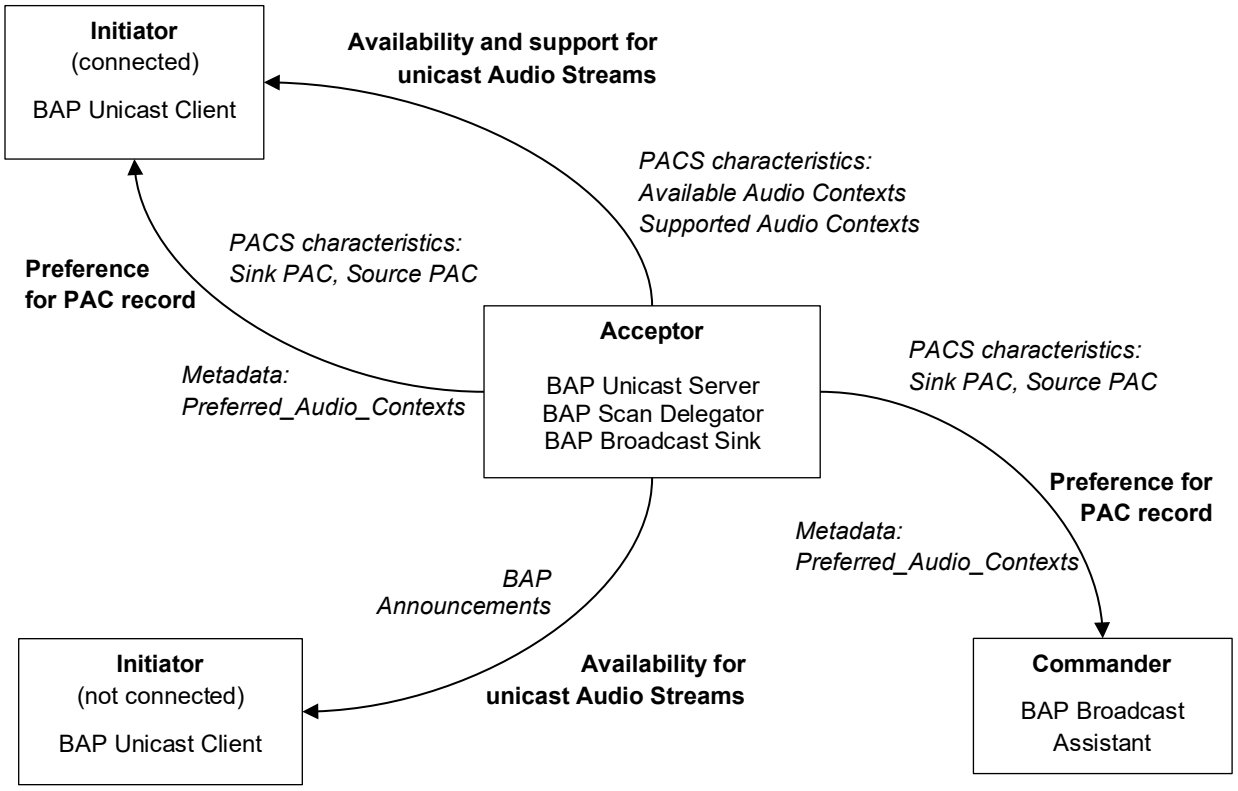

Figure 2.4: Flow of Context Type information as used to signal support, availability, and configuration preferences for Context Type values

3 References

[1] Bluetooth Assigned Numbers, https://www.bluetooth.com/specifications/assigned-numbers

[2] Common Audio Profile Specification, Version 1.0

[3] Media Control Service Specification, Version 1.0

[4] Telephone Bearer Service Specification, Version 1.0

[5] Published Audio Capabilities Service Specification, Version 1.0.1 or later

[6] Basic Audio Profile Specification, Version 1.0.1 or later