Published Audio Capabilities Service

***Bluetooth*® Service Specification**

- **Version:** v1.0.2
- **Version Date:** 2024-10-01
- **Prepared By:** Generic Audio Working Group

Abstract

The Published Audio Capabilities (PACS) service exposes server audio capabilities and audio availability, allowing discovery by clients.

Version History

Version Number	Date	Comments
v1.0	2021-09-14	Adopted by the Bluetooth SIG Board of Directors.
v1.0.1	2022-06-21	Adopted by the Bluetooth SIG Board of Directors.
v1.0.2	2024-10-01	Adopted by the Bluetooth SIG Board of Directors.

Acknowledgments

Name	Company
Jonathan Tanner	Qualcomm Technologies International, Ltd
Chris Church	Qualcomm Technologies International, Ltd
Robin Heydon	Qualcomm Technologies International, Ltd
Nick Hunn	GN Hearing A/S
Søren Larsen	Widex
Markus Schnell	Fraunhofer IIS
Jeff Solum	Starkey
Masahiko Seki	Sony
Andrew Estrada	Sony
Stephan Gehring	Sonova AG
Michael Ungstrup	Widex A/S
Simon Jonghun Song	LG Electronics, Inc.
Bjarne Klemmensen	Oticon A/S
Kanji Kerai	Widex A/S
Erwin Weinans	Plantronics Inc.
Scott Walsh	Plantronics Inc.
Georg Dickmann	Sonova AG
Peter Liu	Bose Corporation
Daniel Sisolak	Bose Corporation
Rasmus Abildgren	Bose Corporation
Xuemei Ouyang	Intel Corporation
Oren Haggai	Intel Corporation
Chethan Narayan Tumkur	Intel Corporation
Siegfried Lehmann	Apple

Name	Company
Riccardo Cavallari	Sivantos GmbH
Marcel Holtmann	Intel Corporation
Sam Geeraerts	NXP Semiconductors
Anil Kumar Vutukuru	MindTree Limited
Luiz Von Dentz	Intel Corporation
Himanshu Bhalla	Intel Corporation
Andrew Credland	Samsung Electronics Co., Ltd
Khaled Elsayed	Synopsys
Michael Rougeux	Bose Corporation
Tim Reilly	Bose Corporation
Ella Chu	Microchip
Charlie Lee	Microchip
Asbjørn Sæbø	Nordic Semiconductor ASA
David Hughes	Broadcom
Sherry Smith	Broadcom
Łukasz Rymanowski	Codecoup
Grzegorz Kołodziejczyk	Codecoup
Morteza Rahchamani	Arm
Frank Yerrace	Microsoft
Dong Jianli	Oppo
Yao Wang	Barrot

Contents

1 Introduction 6

1.1 Conformance 6
1.2 Service dependencies 6
1.3 Bluetooth Core Specification release compatibility 6
1.4 GATT sub-procedure requirements 6
1.5 Transport dependencies 6
1.6 Application error codes 6
1.7 Byte transmission order 7
1.8 Change History 7
1.8.1 Changes from v1.0.1 to v1.0.2 7
1.9 Language 7
1.9.1 Language conventions 7
1.9.2 Reserved for Future Use 8
1.9.3 Prohibited 8
1.10 Terminology 9

2 Service 10

2.1 Declaration 10
2.2 Behavior 10

3 Service characteristics 14

3.1 Sink PAC 14
3.1.1 Sink PAC behavior 16
3.2 Sink Audio Locations 16
3.2.1 Sink Audio Locations behavior 16
3.3 Source PAC 17
3.3.1 Source PAC behavior 18
3.4 Source Audio Locations 19
3.4.1 Source Audio Locations behavior 19
3.5 Available Audio Contexts 20
3.5.1 Available Audio Contexts behavior 20
3.6 Supported Audio Contexts 21
3.6.1 Supported Audio Contexts behavior 21

4 SDP interoperability 23

5 Acronyms and abbreviations 24

6 References 25

1 Introduction

1.1 Conformance

Each capability of this specification shall be supported in the specified manner. This specification may provide options for design flexibility, because, for example, some products do not implement every portion of the specification. For each implementation option that is supported, it shall be supported as specified.

1.2 Service dependencies

This service requires the Generic Attribute Profile (GATT) described in [2]. This service is not dependent on other services.

1.3 Bluetooth Core Specification release compatibility

This service is compatible with Bluetooth Core Specification, Version 5.2 [2] or later.

1.4 GATT sub-procedure requirements

Requirements in this section represent a minimum set of requirements for a server. Other GATT sub-procedures may be used if supported by both the client and server.

Table 1.1 summarizes additional GATT sub-procedure requirements beyond those required by all GATT servers on Unenhanced Attribute Protocol (ATT) bearers.

Requirements in this section are defined as “Mandatory” (M), “Optional” (O), “Excluded” (X), and “Conditional” (C.*n*). Conditional requirements (C.*n*) are listed directly below the table in which they appear.

GATT Sub-Procedure	Requirements
Write Characteristic Value	M
Notifications	M
Read Characteristic Descriptors	M
Write Characteristic Descriptors	M

Table 1.1: Additional GATT sub-procedure requirements on Unenhanced ATT bearers

If the server supports characteristic values larger than the minimum (ATT_MTU -1) for the Unenhanced ATT bearer, then the server should support the Read Long Characteristic Values GATT sub-procedure if not already required by the Bluetooth Core Specification [2].

1.5 Transport dependencies

This specification does not impose any transport requirements. If reliability of Notifications is required (See Volume 3, Part F, Section 3.3.2 in [2]), higher layers can require Enhanced ATT bearer support.

1.6 Application error codes

This service does not define any ATT application error codes.

1.7 Byte transmission order

All characteristics used with this service shall be transmitted with the least significant octet (LSO) first (i.e., little endian). The LSO is identified in the characteristic definitions in Bluetooth Assigned Numbers [1].

1.8 Change History

This section summarizes changes at a moderate level of detail and should not be considered representative of every change made.

1.8.1 Changes from v1.0.1 to v1.0.2

Section	Errata
1.1: Conformance	23862
1.10: Terminology	18854
2.2: Behavior	19094, 22299
3.1: Sink PAC	19104
3.2: Sink Audio Locations	22952
3.3: Source PAC	19104
3.4: Source Audio Locations	22952
3.5: Available Audio Contexts	22298
3.6: Supported Audio Contexts	19232, 22298
6: References	19309

Table 1.2: Errata incorporated in v1.0.2

1.9 Language

1.9.1 Language conventions

The Bluetooth SIG has established the following conventions for use of the words ***shall***, ***must***, ***will***, ***should***, ***may***, ***can***, ***is***, and ***note*** in the development of specifications:

shall	is required to – used to define requirements.
must	is used to express: a natural consequence of a previously stated mandatory requirement. OR an indisputable statement of fact (one that is always true regardless of the circumstances).
will	it is true that – only used in statements of fact.

should	is recommended that – used to indicate that among several possibilities one is recommended as particularly suitable, but not required.
may	is permitted to – used to allow options.
can	is able to – used to relate statements in a causal manner.
is	is defined as – used to further explain elements that are previously required or allowed.
note	Used to indicate text that is included for informational purposes only and is not required in order to implement the specification. Each note is clearly designated as a “Note” and set off in a separate paragraph.

For clarity of the definition of those terms, see Core Specification Volume 1, Part E, Section 1.

1.9.2 Reserved for Future Use

Where a field in a packet, Protocol Data Unit (PDU), or other data structure is described as "Reserved for Future Use" (irrespective of whether in uppercase or lowercase), the device creating the structure shall set its value to zero unless otherwise specified. Any device receiving or interpreting the structure shall ignore that field; in particular, it shall not reject the structure because of the value of the field.

Where a field, parameter, or other variable object can take a range of values, and some values are described as "Reserved for Future Use," a device sending the object shall not set the object to those values. A device receiving an object with such a value should reject it, and any data structure containing it, as being erroneous; however, this does not apply in a context where the object is described as being ignored or it is specified to ignore unrecognized values.

When a field value is a bit field, unassigned bits can be marked as Reserved for Future Use and shall be set to 0. Implementations that receive a message that contains a Reserved for Future Use bit that is set to 1 shall process the message as if that bit was set to 0, except where specified otherwise.

The acronym RFU is equivalent to Reserved for Future Use.

1.9.3 Prohibited

When a field value is an enumeration, unassigned values can be marked as “Prohibited.” These values shall never be used by an implementation, and any message received that includes a Prohibited value shall be ignored and shall not be processed and shall not be responded to.

Where a field, parameter, or other variable object can take a range of values, and some values are described as “Prohibited,” devices shall not set the object to any of those Prohibited values. A device receiving an object with such a value should reject it, and any data structure containing it, as being erroneous.

“Prohibited” is never abbreviated.

1.10 Terminology

Table 1.3 defines terms that are needed to understand features used in this service. This service also uses terms that are defined in the Basic Audio Profile (BAP) [3].

Term	Definition
Audio Location	Defined in [3].
broadcast Audio Stream	Defined in [3].
Context Type	A bitfield of values that, when set to 0b1 for a bit, describes audio data as being intended for the use case represented by that bit. Context Type values are defined in Bluetooth Assigned Numbers [1].
EATT	An ATT bearer feature introduced in Volume 3, Part F, Section 3.2.11 in [2]
Enhanced ATT bearer	An ATT bearer using the Enhanced Credit Based Flow Control Logical Link Control and Adaptation Protocol (L2CAP) channel mode introduced in Volume 3, Part A, Section 10.2 in [2]
Metadata	Defined in [3]
PAC record	A set of parameter values that denote server audio capabilities.
reliability	Defined in Volume 3, Part F, Section 3.3.2 in [2]
Supported_-Sampling_Frequencies	Defined in [3]
Unenhanced ATT bearer	An ATT bearer not using the Enhanced Credit Based Flow Control L2CAP channel mode introduced in Volume 3, Part A, Section 10.2 in [2]
unicast Audio Stream	Defined in [3]

Table 1.3: Terminology

2 Service

2.1 Declaration

There shall be no more than one instance of the Published Audio Capabilities Service (PACS) on a server.

PACS should be a «Primary Service» and the service universally unique identifier (UUID) shall be set to «Published Audio Capabilities» as defined in [1].

2.2 Behavior

PACS can be instantiated on devices that can accept the establishment of unicast Audio Streams or devices that can receive broadcast Audio Streams. Examples of such devices are speakers, headsets, hearing aids, and microphones.

Servers expose one or more sets of audio capabilities and audio availability. Sets of audio capabilities, known as Published Audio Capability (PAC) records, are exposed by using either the Sink PAC characteristic or Source PAC characteristic. Clients can discover and read these characteristics, and servers can notify these characteristics.

Audio availability is exposed using the Available Audio Contexts characteristic. Clients can discover and read this characteristic, and servers can notify this characteristic. Audio availability represents the server's determination of whether it considers itself available to accept the establishment of a unicast Audio Stream based on classifications of the use cases intended for such Audio Streams.

Audio capabilities, exposed in PAC records, represent the server audio capabilities independent of available resources at any given time. Audio capabilities do not distinguish between unicast Audio Streams or broadcast Audio Streams.

The server may expose multiple supported values for a given parameter in a PAC record where the format of that parameter value allows, such as bitfields or ranges.

When exposing support for multiple values in a parameter within a PAC record, the server shall support all possible combinations of that parameter value with the other parameter values contained in the same PAC record (e.g., whether single values, bitfields, or ranges).

For example, consider a server that exposes a PAC record where the server supports reception of audio data using the Sink PAC characteristic. The format of the parameters used in Table 2.1 is described in Section 3.1 for the Sink PAC characteristic.

PAC Record	**i**
Codec_ID	0x000000000D
Codec_Specific_Capabilities_Length	0x0A

PAC Record		**i**
Codec_Specific_Capabilities		0x0032001E040500060103
	Length	0x03
	Type: Supported_Sampling_Frequencies	0x01
	Value	0x0006
	Length	0x05
	Type: Supported_Octets_Per_Codec_Frame	0x04
	Value	0x0032001E
Metadata_Length		0x00
Metadata		(empty)

Table 2.1: Example of a PAC record where the server supports reception of audio data

The Supported_Sampling_Frequencies [3] parameter value is formatted as a bitfield and has more than one value supported in the PAC record which is labelled i in Table 2.1. The value of 0x0006 represents the bit-wise OR of the individual bits 0x0002 and 0x0004, which means the server was exposing support for the sampling frequencies 16000 Hz and 24000 Hz.

The Supported_Octets_Per_Codec_Frame [3] parameter value is formatted as a range of all possible values from the minimum supported octets per codec frame to the maximum supported octets per codec frame. The value of 0x0032001E means the server was exposing support for a range of 30 (0x001E) to 50 (0x0032) octets per codec frame.

This example use of these values means the server was exposing support for every combination of the two Supported_Sampling_Frequencies and all 21 possible values within the Supported_Octets_Per_Codec_Frame range.

PAC record i would be represented, when expanded, as shown in Table 2.1.

PAC Record		**i (set 1)**	**i (set 2)**
Codec_ID		0x000000000D	0x000000000D
Codec_Specific_Capabilities_Length		0x0A	0x0A
Codec_Specific_Capabilities		0x0032001E040500020103	0x0032001E040500040103
	Length	0x03	0x03

PAC Record		**i (set 1)**	**i (set 2)**
	Type: Supported_Sampling_Frequencies	0x01	0x01
	Value	0x0002	0x0004
	Length	0x05	0x05
	Type: Supported_Octets_Per_Codec_Frame	0x04	0x04
	Value	0x0032001E	0x0032001E
Metadata_Length		0x00	0x00
	Metadata	(empty)	(empty)

Table 2.2: Representation of expanded PAC record in Table 2.1

A client reading PAC record i might request a Codec_Specific_Configuration that included an Octets_Per_Codec_Frame parameter value set to any number between 30 to 50 octets inclusive.

If the server wished to support only 30-octet codec frame lengths and 50-octet codec frame lengths, but not support codec frame lengths in-between these minimum and maximum values, then the server would need to expose discrete PAC records with the minimum and maximum values set to 30 octets in one PAC record, and with the minimum and maximum values set to 50 octets in another PAC record, as shown in Table 2.3.

PAC Record		**i**	**j**
Codec_ID		0x000000000D	0x000000000D
Codec_Specific_Capabilities_Length		0x0A	0x0A
Codec_Specific_Capabilities		0x001E001E040500060103	0x00320032040500060103
	Length	0x03	0x03
	Type: Supported_Sampling_Frequencies	0x01	0x01
	Value	0x0006	0x0006
	Length	0x05	0x05
	Type: Supported_Octets_Per_Codec_Frame	0x04	0x04
	Value	0x001E001E	0x00320032
Metadata_Length		0x00	0x00
Metadata		(empty)	(empty)

Table 2.3: Splitting the PAC record from Table 2.1 to expose discrete values for parameters formatted as ranges

PAC records i and j in Table 2.3, when expanded, would be represented as shown in Table 2.4.

PAC Record		i (set 1)	i (set 2)	j (set 1)	j (set 2)
Codec_ID		0x000000000D	0x000000000D	0x000000000D	0x000000000D
Codec_Specific_Capabilities_Length		0x0A	0x0A	0x0A	0x0A
Codec_Specific_Capabilities		0x001E001E040500020103	0x001E001E0405000040103	0x003200320405040500020103	0x003200320405000040103
	Length	0x03	0x03	0x03	0x03
	Type: Supported_Sampling_Frequencies	0x01	0x01	0x01	0x01
	Value	0x0002	0x0004	0x0002	0x0004
	Length	0x05	0x05	0x05	0x05
	Type: Supported_Octets_Per_Codec_Frame	0x04	0x04	0x04	0x04
	Value	0x001E001E	0x001E001E	0x00320032	0x00320032
Metadata_Length		0x00	0x00	0x00	0x00
	Metadata	(empty)	(empty)	(empty)	(empty)

Table 2.4: Representing PAC records i and j in expanded form

For all characteristics defined in this specification, arrayed parameters are specified by using the following notation: ParameterA[i]. If more than one set of arrayed parameters is specified (e.g., ParameterA[i], ParameterB[i]), then the order of the parameters is as follows (unless noted otherwise): ParameterA[0], ParameterB[0], ParameterA[1], ParameterB[1], ParameterA[2], ParameterB[2], … ParameterA[n], ParameterB[n].

3 Service characteristics

This section defines the characteristic and descriptor requirements for PACS. The characteristics are shown in Table 3.1.

Requirements in this section are defined as “Mandatory” (M), “Optional” (O), “Excluded” (X), and “Conditional” (C.*n*). Conditional requirements (C.*n*) are listed directly below the table in which they appear.

Characteristic Name	Requirement	Mandatory Properties	Optional Properties	Security Permissions
Sink PAC	C.1	Read	Notify	Encryption required
Sink Audio Locations	C.2	Read	Notify, Write	Encryption required
Source PAC	C.1	Read	Notify	Encryption required
Source Audio Locations	C.3	Read	Notify, Write	Encryption required
Available Audio Contexts	M	Read, Notify	None	Encryption required
Supported Audio Contexts	M	Read	Notify	Encryption required

Table 3.1: PACS characteristics

C.1: Mandatory to support at least one of the Sink PAC characteristic or Source PAC characteristic.

C.2: Optional to support if the Sink PAC characteristic is supported, otherwise Excluded.

C.3: Optional to support if the Source PAC characteristic is supported, otherwise Excluded.

At least one Sink PAC characteristic shall exist on the server if the Sink PAC characteristic is supported.

A single Sink Audio Locations characteristic may exist on the server if the Sink PAC characteristic is supported.

At least one Source PAC characteristic shall exist on the server if the Source PAC characteristic is supported.

A single Source Audio Locations characteristic may exist on the server if the Source PAC characteristic is supported.

A single Available Audio Contexts characteristic shall exist on the server.

A single Supported Audio Contexts characteristic shall exist on the server.

3.1 Sink PAC

The Sink PAC characteristic is used to expose PAC records when the server supports reception of audio data.

When the server supports reception of audio data, the decision to expose all supported PAC records within a single Sink PAC characteristic or multiple Sink PAC characteristics is left to the implementation.

Examples of situations where a server that supports reception of audio data might want to expose multiple Sink PAC characteristics:

- The server wanted to support a smaller maximum transmission unit (ATT_MTU, as defined in Volume 3, Part F, Section 3.2.8 in [2]) size. Exposing all supported PAC records in a single Sink PAC characteristic would require the server to increase its supported Maximum Transmission Unit (MTU) size to a value the server considered excessive.
- The server wanted to expose support for proprietary audio capabilities (such as vendor-specific audio codecs, as denoted by the Codec_ID parameter value) separately from support for non- vendor-specific audio capabilities and used separate Sink PAC characteristics to expose such support.
- The server wanted to minimize the amount of data to be transferred, when sending notifications to a client that the Sink PAC characteristic value changed, by exposing the audio capabilities likely to change quicker than others in separate Sink PAC characteristics.

The Sink PAC characteristic format is defined in Table 3.2.

Parameter	Size (Octets)	Description
Number_of_PAC_records	1	Number of PAC records, [i], in this characteristic. Shall be ≥ 1.
Codec_ID[i]	5	Octet 0: Coding Format value of the [ith] PAC record. Coding Format values are defined in the Host Controller Interface section of Bluetooth Assigned Numbers [1]. Octet 1–2: Company_ID value of the [ith] PAC record. Shall be 0x0000 if octet 0 is not 0xFF. Company_ID values are defined in Bluetooth Assigned Numbers [1]. Octet 3–4: Vendor-specific codec_ID value of the [ith] PAC record. Shall be 0x0000 if octet 0 is not 0xFF.
Codec_Specific_-Capabilities_Length[i]	1	Length, in octets, of the Codec_Specific_Capabilities value of the [ith] PAC record. Shall be 0x00 if the Codec_Specific_Capabilities value of the [ith] PAC record is empty.
Codec_Specific_-Capabilities[i]	Varies	Codec_Specific_Capabilities value of the [ith] PAC record.
Metadata_Length[i]	1	Length of the Metadata field of the [ith] PAC record. Shall be 0x00 if the Metadata value of the [ith] PAC record value is empty.
Metadata[i]	Varies	Length-type-value (LTV)-formatted Metadata applicable to the [ith] PAC record. Shall exist only if the value of the Metadata_Length field is not 0x00.

Table 3.2: Format of Sink PAC characteristic

3.1.1 Sink PAC behavior

The Sink PAC characteristic returns its characteristic value when read by a client.

If the server supports changing a PAC record, the server shall support notifications for the Sink PAC characteristic instance containing the PAC record that can change, e.g., if the server has updated firmware, or if a new application supporting different codecs or different capabilities for currently supported codecs is installed, then the characteristic value could be updated.

If notifications are supported, the Sink PAC characteristic can be configured for notifications by using the GATT *Write Characteristic Descriptors* sub-procedure on the Client Characteristic Configuration descriptor.

If the characteristic value changes when in a connection, and the value of the Client Characteristic Configuration descriptor is configured for notifications, the server shall notify the new characteristic value to the client.

If the characteristic value changes when not in a connection, and the value of the Client Characteristic Configuration descriptor is configured for notifications, the server shall notify the new characteristic value when reconnecting to a bonded client.

3.2 Sink Audio Locations

The Sink Audio Locations characteristic is used to expose the supported Audio Locations when the server supports reception of audio data.

The Sink Audio Locations characteristic format is defined in Table 3.3.

Parameter	Size (Octets)	Description
Sink_Audio_Locations	4	Device-wide bitmap of supported Audio Location values for all PAC records where the server supports reception of audio data. Audio Location values are defined in Bluetooth Assigned Numbers [1]. The value 0x00000000 indicates that the server supports receiving only mono audio (no specified Audio Location). For all other values, the identified locations and mono are supported.

Table 3.3: Format of Sink Audio Locations characteristic

3.2.1 Sink Audio Locations behavior

The Sink Audio Locations characteristic returns its characteristic value when read by a client.

If the server supports local changes to the value of the Sink Audio Locations characteristic, the server shall support notifications of the Sink Audio Locations characteristic.

The server may support writes to the Sink Audio Locations characteristic value by clients.

If the server supports writes to the Sink Audio Locations characteristic by clients, the server shall support notifications of the Sink Audio Locations characteristic.

If writes are supported, the Sink Audio Locations characteristic can be written by a client.

If notifications are supported, the characteristic can be configured for notifications by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor.

If the characteristic value changes when in a connection, and the value of the Client Characteristic Configuration descriptor is configured for notifications, the server shall notify the new characteristic value to the client.

If the characteristic value changes when not in a connection, and the value of the Client Characteristic Configuration descriptor is configured for notifications, the server shall notify the new characteristic value when reconnecting to a bonded client.

3.2.1.1 Error Handling

If the server detects that the Sink_Audio_Locations parameter value, written by a client by using the GATT Write Characteristic Value sub-procedure, is not 4 octets in length, or if the parameter value written includes any RFU bits set to a value of 0b1, the server shall respond with an ATT Error Response and shall set the Error Code parameter to Write Request Rejected as defined in [4].

3.3 Source PAC

The Source PAC characteristic is used to expose PAC records when the server supports transmission of audio data.

When the server supports transmission of audio data, the decision to expose all supported PAC records within a single Source PAC characteristic or multiple Source PAC characteristics is left to the implementation.

Examples of situations where a server that supports transmission of audio data might want to expose multiple Source PAC characteristics:

- The server wanted to support a smaller maximum transmission unit (ATT_MTU, as defined in Volume 3, Part F, Section 3.2.8 in [2]) size. Exposing all supported PAC records in a single Source PAC characteristic would require the server to increase its supported MTU size to a value the server considered excessive.
- The server wanted to expose support for proprietary audio capabilities (such as vendor-specific audio codecs, as denoted by the Codec_ID parameter value) separately from support for non-vendor-specific audio capabilities and used separate Source PAC characteristics to expose such support.
- The server wanted to minimize the amount of data to be transferred, when sending notifications to a client that the Source PAC characteristic value changed, by exposing the audio capabilities likely to change quicker than others in separate Source PAC characteristics.

The Source PAC characteristic format is defined in Table 3.4.

Parameter	Size (Octets)	Description
Number_of_PAC_records	1	Number of PAC records, [i], for this characteristic. Shall be ≥ 1.
Codec_ID[i]	5	Octet 0: Coding Format value of the [ith] PAC record. Coding Format values are defined in the Host Controller Interface section of Bluetooth Assigned Numbers [1]. Octet 1–2: Company_ID value of the [ith] PAC record. Shall be 0x0000 if octet 0 is not 0xFF. Company_ID values are defined in Bluetooth Assigned Numbers [1]. Octet 3–4: Vendor-specific codec_ID value of the [ith] PAC record. Shall be 0x0000 if octet 0 is not 0xFF.
Codec_Specific_-Capabilities_Length[i]	1	Length of the Codec_Specific_Capabilities value of the [ith] PAC record. Shall be 0x00 if the Codec_Specific_Capabilities value of the [ith] PAC record is empty.
Codec_Specific_-Capabilities[i]	*Varies*	Codec_Specific_Capabilities value of the [ith] PAC record.
Metadata_Length[i]	1	Length of the Metadata field of the [ith] PAC record. Shall be 0x00 if the Metadata value of the [ith] PAC record value is empty.
Metadata[i]	Varies	LTV-formatted Metadata applicable to the [ith] PAC record. Shall exist only if the value of the Metadata_Length field is not 0x00.

Table 3.4: Format of Source PAC characteristic

3.3.1 Source PAC behavior

The Source PAC characteristic returns its characteristic value when read by a client. The characteristic can be configured for notifications by using the GATT *Write Characteristic Descriptors* sub-procedure on the Client Characteristic Configuration descriptor if the server supports notifications of the Source PAC characteristic.

If the server supports changing a PAC record, the server shall support notifications for the Source PAC characteristic instance containing the PAC record that can change, e.g., if the server has updated firmware, or if a new application supporting different codecs or different capabilities for currently supported codecs is installed, then the characteristic value could be updated.

If the characteristic value changes when in a connection, and the value of the Client Characteristic Configuration descriptor is configured for notifications, the server shall notify the new characteristic value to the client.

If the characteristic value changes when not in a connection, and the value of the Client Characteristic Configuration descriptor is configured for notifications, the server shall notify the new characteristic value when reconnecting to a bonded client.

3.4 Source Audio Locations

The Source Audio Locations characteristic is used to expose the supported Audio Locations when the server supports transmission of audio data.

The Source Audio Locations characteristic format is defined in Table 3.5.

Parameter	Size (Octets)	Description
Source_Audio_Locations	4	Device-wide bitmap of supported Audio Location values for PAC records where the server supports transmission of audio data. Audio Location values are defined in Bluetooth Assigned Numbers [1]. The value 0x00000000 indicates that the server supports transmitting only mono audio (no specified Audio Location). For all other values, the identified locations and mono are supported.

Table 3.5: Format of Source Audio Locations characteristic

3.4.1 Source Audio Locations behavior

The Source Audio Locations characteristic returns its characteristic value when read by a client.

If the server supports local changes to the value of the Source Audio Locations characteristic, the server shall support notifications of the Source Audio Locations characteristic.

The server may support writes to the Source Audio Locations characteristic value by clients.

If the server supports writes to the Source Audio Locations characteristic by clients, the server shall support notifications of the Source Audio Locations characteristic.

If writes are supported, the Source Audio Locations characteristic can be written by a client.

If notifications are supported, the characteristic can be configured for notifications by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor.

If the characteristic value changes when in a connection, and the value of the Client Characteristic Configuration descriptor is configured for notifications, the server shall notify the new characteristic value to the client.

If the characteristic value changes when not in a connection, and the value of the Client Characteristic Configuration descriptor is configured for notifications, the server shall notify the new characteristic value when reconnecting to a bonded client.

3.4.1.1 Error Handling

If the server detects that the Source_Audio_Locations parameter value, written by a client by using the GATT Write Characteristic Value sub-procedure, is not 4 octets in length, or if the parameter value written

includes any RFU bits set to a value of 0b1, the server shall respond with an ATT Error Response and shall set the Error Code parameter to Write Request Rejected as defined in [4].

3.5 Available Audio Contexts

The Available Audio Contexts characteristic exposes the availability of the server for reception and/or transmission of unicast audio data only, associated with specific Context Types.

The determination of whether the server considers itself available is left to the implementation, unless defined by higher-layer specifications.

The Available Audio Contexts characteristic format is defined in Table 3.6.

Field	Size (Octets)	Description
Available_Sink_Contexts	2	Bitmask of audio data Context Type values available for reception. 0x0000 = server is not available to receive audio for any Context Type value. Context Type values are defined in Bluetooth Assigned Numbers [1].
Available_Source_Contexts	2	Bitmask of audio data Context Type values available for transmission. 0x0000 = server is not available to transmit audio for any Context Type values. Context Type values are defined in Bluetooth Assigned Numbers [1].

Table 3.6: Format of Available Audio Contexts characteristic

3.5.1 Available Audio Contexts behavior

The Available Audio Contexts characteristic returns its characteristic value when read by a client.

The characteristic can be configured for notifications by clients by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor.

The server may expose different values of the Available Audio Contexts characteristic to each client.

If the corresponding bit in the Supported Audio Contexts characteristic value is not set to 0b1, the server shall not set a bit to 0b1 in the Available Audio Contexts characteristic value.

The server shall update the value of Available_Sink_Contexts and/or Available_Source_Contexts fields when a change in its availability occurs, as determined by the implementation or by a higher-layer specification.

If the characteristic value exposed for a client changes when in a connection, and the value of the Client Characteristic Configuration descriptor is configured for notifications, the server shall notify the new characteristic value to the client.

If the characteristic value exposed for a client changes when not in a connection, and the value of the Client Characteristic Configuration descriptor is configured for notifications, the server shall notify the new characteristic value when reconnecting to a bonded client.

3.6 Supported Audio Contexts

The Supported Audio Contexts characteristic exposes the server's support for reception and/or transmission of unicast audio data and/or broadcast audio data associated with specific Context Types.

The value of the Supported Audio Contexts characteristic allows clients to determine whether the server supports a particular Context Type, which is distinct from the server's availability to receive and/or transmit audio data associated with that Context Type as defined by the value of the Available Audio Contexts characteristic.

The Supported Audio Contexts characteristic format is defined in Table 3.7.

Field	Size (Octets)	Description
Supported_Sink_Contexts	2	Bitmask of audio data Context Type values supported for reception. Shall be 0x0000 if the server does not support the Sink PAC characteristic. Context Type values are defined in Bluetooth Assigned Numbers [1].
Supported_Source_Contexts	2	Bitmask of audio data Context Type values supported for transmission. Shall be 0x0000 if the server does not support the Source PAC characteristic. Context Type values are defined in Bluetooth Assigned Numbers [1].

Table 3.7: Format of Supported Audio Contexts characteristic

3.6.1 Supported Audio Contexts behavior

The Supported Audio Contexts characteristic returns its characteristic value when read by a client.

If the server supports changes to its supported audio data Context Types, the server shall support notifications of the Supported Audio Contexts characteristic. The server shall update the value of Supported_Sink_Contexts and/or Supported_Source_Contexts fields if a change in its supported Context Types occurs, for example, as the result of a software update.

If notifications are supported, the Supported Audio Contexts characteristic can be configured for notifications by clients by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor.

If the characteristic value changes when in a connection, and the value of the Client Characteristic Configuration descriptor is configured for notifications, the server shall notify the new characteristic value to the client.

If the characteristic value changes when not in a connection, and the value of the Client Characteristic Configuration descriptor is configured for notifications, the server shall notify the new characteristic value when reconnecting to a bonded client.

4 SDP interoperability

When PACS is supported over Basic Rate/Enhanced Data Rate (BR/EDR), the attributes defined in Table 4.1 shall be included in the Service Discovery Protocol (SDP) service record.

Item	Definition	Type	Value	Requirement
Service Class ID List	–	–	–	M
Service Class #0	–	UUID	«Published Audio Capabilities»	M
Protocol Descriptor List	–	Data Element Sequence	–	M
Protocol #0	–	UUID	«L2CAP»	M
Parameter #0 for Protocol #0	Protocol/Service Multiplexer (PSM)	uint16	PSM = ATT	M
Protocol #1	–	UUID	«ATT»	M
Additional Protocol Descriptor List	–	Data Element Sequence	–	C.1
Protocol Descriptor List	–	Data Element Sequence	–	C.1
Protocol #0	–	UUID	«L2CAP»	C.1
Parameter #0 for Protocol #0	PSM	uint16	PSM = EATT	C.1
Protocol #1	–	UUID	«ATT»	C.1
BrowseGroupList	–	–	PublicBrowseRoot Other browse UUIDs may also be included in the list.	M

Table 4.1: SDP record

C.1: Mandatory to support if EATT is supported, otherwise Excluded.

5 Acronyms and abbreviations

Acronym/Abbreviation	Meaning
ATT	Attribute Protocol
BAP	Basic Audio Profile
BR/EDR	Basic Rate/Enhanced Data Rate
EATT	Enhanced Attribute Protocol
GATT	Generic Attribute Profile
L2CAP	Logical Link Control and Adaptation Protocol
LSO	least significant octet
MTU	Maximum Transmission Unit
PAC	Published Audio Capability
PACS	Published Audio Capabilities Service
PDU	Protocol Data Unit
PSM	Protocol Service Multiplexer
RFU	Reserved for Future Use
SDP	Service Discovery Protocol
UUID	universally unique identifier

Table 5.1: Acronyms and abbreviations

6 References

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

[2] Bluetooth Core Specification, Version 5.2 or later

[3] Basic Audio Profile Specification, Version 1.0 or later

[4] Bluetooth Core Specification Supplement, Version 11 or later