

Broadcast Audio Scan Service

Bluetooth® Service Specification

- **Version:** v1.0.1
- **Version Date:** 2025-02-11
- **Prepared By:** Generic Audio Working Group

Abstract

This service is used by servers to expose their status with respect to synchronization to broadcast Audio Streams and associated data, including Broadcast_Codes used to decrypt encrypted broadcast Audio Streams. Clients can use the attributes exposed by servers to observe and/or request changes in server behavior.



Version History

Version Number	Date	Comments
v1.0	2021-09-14	Adopted by the Bluetooth SIG Board of Directors.
v1.0.1	2025-02-11	Adopted by the Bluetooth SIG Board of Directors.

Acknowledgments

Name	Company
Robin Heydon	Qualcomm
Jonathan Tanner	Qualcomm
Chris Church	Qualcomm
Jeff Solum	Starkey
Nick Hunn	GN Resound
Rasmus Abildgren	Bose
Bjarne Klemmensen	Oticon A/S
Søren Larsen	Widex
Markus Schnell	Fraunhofer IIS
Masahiko Seki	Sony
Andrew Estrada	Sony
Stephan Gehring	Sonova AG
Michael Ungstrup	Widex A/S
Simon Jonghun Song	LG Electronics, Inc.
HJ Lee	LG Electronics, Inc.
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
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

Use of this specification is your acknowledgement that you agree to and will comply with the following notices and disclaimers. You are advised to seek appropriate legal, engineering, and other professional advice regarding the use, interpretation, and effect of this specification.

Use of Bluetooth specifications by members of Bluetooth SIG is governed by the membership and other related agreements between Bluetooth SIG and its members, including those agreements posted on Bluetooth SIG's website located at www.bluetooth.com. Any use of this specification by a member that is not in compliance with the applicable membership and other related agreements is prohibited and, among other things, may result in (i) termination of the applicable agreements and (ii) liability for infringement of the intellectual property rights of Bluetooth SIG and its members. This specification may provide options, because, for example, some products do not implement every portion of the specification. All content within the specification, including notes, appendices, figures, tables, message sequence charts, examples, sample data, and each option identified is intended to be within the bounds of the Scope as defined in the Bluetooth Patent/Copyright License Agreement ("PCLA"). Also, the identification of options for implementing a portion of the specification is intended to provide design flexibility without establishing, for purposes of the PCLA, that any of these options is a "technically reasonable non-infringing alternative."

Use of this specification by anyone who is not a member of Bluetooth SIG is prohibited and is an infringement of the intellectual property rights of Bluetooth SIG and its members. The furnishing of this specification does not grant any license to any intellectual property of Bluetooth SIG or its members. THIS SPECIFICATION IS PROVIDED "AS IS" AND BLUETOOTH SIG, ITS MEMBERS AND THEIR AFFILIATES MAKE NO REPRESENTATIONS OR WARRANTIES AND DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, TITLE, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR THAT THE CONTENT OF THIS SPECIFICATION IS FREE OF ERRORS. For the avoidance of doubt, Bluetooth SIG has not made any search or investigation as to third parties that may claim rights in or to any specifications or any intellectual property that may be required to implement any specifications and it disclaims any obligation or duty to do so.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, BLUETOOTH SIG, ITS MEMBERS AND THEIR AFFILIATES DISCLAIM ALL LIABILITY ARISING OUT OF OR RELATING TO USE OF THIS SPECIFICATION AND ANY INFORMATION CONTAINED IN THIS SPECIFICATION, INCLUDING LOST REVENUE, PROFITS, DATA OR PROGRAMS, OR BUSINESS INTERRUPTION, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, AND EVEN IF BLUETOOTH SIG, ITS MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF THE DAMAGES.

Products equipped with Bluetooth wireless technology ("Bluetooth Products") and their combination, operation, use, implementation, and distribution may be subject to regulatory controls under the laws and regulations of numerous countries that regulate products that use wireless non-licensed spectrum. Examples include airline regulations, telecommunications regulations, technology transfer controls, and health and safety regulations. You are solely responsible for complying with all applicable laws and regulations and for obtaining any and all required authorizations, permits, or licenses in connection with your use of this specification and development, manufacture, and distribution of Bluetooth Products. Nothing in this specification provides any information or assistance in connection with complying with applicable laws or regulations or obtaining required authorizations, permits, or licenses.

Bluetooth SIG is not required to adopt any specification or portion thereof. If this specification is not the final version adopted by Bluetooth SIG's Board of Directors, it may not be adopted. Any specification adopted by Bluetooth SIG's Board of Directors may be withdrawn, replaced, or modified at any time. Bluetooth SIG reserves the right to change or alter final specifications in accordance with its membership and operating agreements.

Copyright © 2019–2025. All copyrights in the Bluetooth Specifications themselves are owned by Apple Inc., Ericsson AB, Intel Corporation, Google LLC, Lenovo (Singapore) Pte. Ltd., Microsoft Corporation, Nokia Corporation, and Toshiba Corporation. The Bluetooth word mark and logos are owned by Bluetooth SIG, Inc. Other third-party brands and names are the property of their respective owners.

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	7
1.7	Byte transmission order	7
1.8	Change History	7
1.8.1	Changes from v1.0 to v1.0.1	7
1.9	Language	8
1.9.1	Language conventions	8
1.9.2	Reserved for Future Use	8
1.9.3	Prohibited	9
1.10	Terminology	9
2	Service	11
2.1	Declaration	11
2.2	Behavior	11
3	Service characteristics	13
3.1	Broadcast Audio Scan Control Point	13
3.1.1	Characteristic behavior	14
3.1.1.1	Error Handling	14
3.1.1.2	Remote Scan Stopped operation	15
3.1.1.3	Remote Scan Started operation	15
3.1.1.4	Add Source operation	15
3.1.1.5	Modify Source operation	19
3.1.1.6	Set Broadcast_Code operation	21
3.1.1.7	Remove Source operation	22
3.2	Broadcast Receive State	22
3.2.1	Characteristic behavior	24
3.2.1.1	Source_ID field	24
3.2.1.2	Source_Address_Type field	25
3.2.1.3	Source_Address field	25
3.2.1.4	Source_Adv_SID field	25
3.2.1.5	Broadcast_ID field	25
3.2.1.6	PA_Sync_State field	26
3.2.1.7	BIG_Encryption field	26
3.2.1.8	Num_Subgroups field	27
3.2.1.9	BIS_Sync_State field	27
3.2.1.10	Metadata_Length field	27
3.2.1.11	Metadata field	28
4	SDP interoperability	29
5	Acronyms and abbreviations	30
6	References	31



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 is not dependent upon 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 Generic Attribute Profile (GATT) sub-procedures may be used if supported by both client and server.

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

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

GATT Sub-Procedure	Requirements
Write Characteristic Value	M
Write Without Response	M
Notifications	M
Read Characteristic Descriptors	M
Write Characteristic Descriptors	M
Write Long Characteristic Value	C.1

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

C.1: Mandatory if the Add Source operation defined in Section 3.1 is supported.

If the server supports characteristic values larger than the minimum ATT_MTU 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 defines the ATT Application error codes shown in [Table 1.2](#).

Name	Error Code	Description
Opcode Not Supported	0x80	An unsupported opcode has been used in a Broadcast Audio Scan Control Point operation.
Invalid Source_ID	0x81	The Source_ID written by a client does not match any Source_ID exposed in a Broadcast Receive State characteristic value by the server.

Table 1.2: 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 [\[1\]](#).

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 to v1.0.1

Section	Errata
1.1: Conformance	23773
3.1.1.4: Add Source operation	26529
3.1.1.5: Modify Source operation	26529
3.2: Broadcast Receive State	26529
3.2.1.7: BIG_Encryption field	22576, 26529
6: References	19297

Table 1.3: Errata incorporated in v1.0.1

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	<u>is required to</u> – 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	<u>it is true that</u> – only used in statements of fact.
should	<u>is recommended that</u> – used to indicate that among several possibilities one is recommended as particularly suitable, but not required.
may	<u>is permitted to</u> – used to allow options.
can	<u>is able to</u> – used to relate statements in a causal manner.
is	<u>is defined as</u> – 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 1.4 defines terms that are needed to understand features used in this specification. This specification also uses terms that are defined in the Basic Audio Profile (BAP) Specification [3] and in the Core Specification [2].

Term	Definition
broadcast Audio Stream	Defined in BAP [3]
Broadcast_ID	Defined in [3]
Broadcast Isochronous Group (BIG)	Defined in Volume 6, Part B, Section 4.4.6.2 in the Bluetooth Core Specification [2]
Broadcast Isochronous Stream (BIS)	Defined in Volume 6, Part B, Section 4.4.6.1 in [2]
Broadcast Source	Defined in [3]
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 Adaption Protocol (L2CAP) channel mode introduced in Volume 3, Part A, Section 10.2 in [2]
Link Layer (LL)	Defined in Volume 6, Part B in [2]
PA_Interval	Corresponds to the SyncInfo field Interval parameter defined in Volume 6, Part B, Section 2.3.4.6 in [2]
Periodic Advertising Synchronization Transfer (PAST) procedure	Defined in Volume 3, Part C, Section 9.5.4 in [2]
periodic advertising train (PA)	Defined in Volume 6, Part B, Section 4.4.5.1 in [2]
Remote Scanning	Defined in [3]

Term	Definition
Service Data AD data type	Defined in the Core Specification Supplement (CSS) [4]
SyncInfo	Defined in Volume 6, Part B, Section 2.3.4.6 in [2]
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]

Table 1.4: Terminology

2 Service

2.1 Declaration

There shall be no more than one Broadcast Audio Scan Service (BASS) instance on a server.

BASS shall be a «Primary Service» and the service universally unique identifier (UUID) shall be set to «Broadcast Audio Scan» as defined in [1].

2.2 Behavior

BASS can be instantiated on servers to solicit for clients to scan on behalf of the server for broadcast Audio Streams and associated data that are transmitted by Broadcast Sources. Clients scanning on behalf of the server can help reduce the need to scan by the server and reduce power consumption on the server.

Servers can receive information from clients that is associated with broadcast Audio Streams, including decryption keys known as Broadcast_Codes (as defined in Volume 3, Part C, Section 3.2.6 in [2]) necessary to decrypt encrypted BISes.

Examples of typical devices that might instantiate BASS include, but are not limited to, hearing aids, headsets, and similar devices with limited battery capacity relative to devices such as televisions, smartphones, and smart watches.

There are two characteristic types used in BASS:

1. A single Broadcast Audio Scan Control Point characteristic that can be used by all clients to:
 - Inform the server whether the client is performing Remote Scanning on behalf of the server.
 - Provide a server with information regarding a Broadcast Source.
 - Request to update and/or remove information regarding a Broadcast Source.
 - Request the server to synchronize to, or stop synchronization to, a PA and/or a BIG containing one or more subgroups (as defined in Section 3.7.2.2 in [3]) containing one or more BISes.
 - Provide broadcast Audio Stream encryption keys (Broadcast_Codes) to the server.
2. One or more Broadcast Receive State characteristics that can be used by all clients to:
 - Determine the server's status with respect to synchronization to one or more PAs and/or a BIG containing one or more subgroups containing one or more BISes.
 - Determine whether the server requires a transfer of SyncInfo data.
 - Determine whether the server is decrypting an encrypted broadcast Audio Stream.
 - Determine whether the server requires a Broadcast_Code to decrypt an encrypted broadcast Audio Stream.

The decision of the server whether to accept an operation written by a client to the Broadcast Audio Scan Control Point characteristic is left to the implementation unless defined by higher layers.

Each Broadcast Receive State characteristic value represents the current state of the server with respect to any Broadcast Source identified in the characteristic value.

The BASS UUID should be included in the Service Data AD Type in advertising data transported using connectable extended advertising packets transmitted by servers supporting this specification. Higher-

layer specifications may define additional service data to be included in the Service Data AD Type used in advertising data.

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.

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

Characteristic Name	Requirement	Mandatory Properties	Optional Properties	Security Permissions
Broadcast Audio Scan Control Point	M	Write, Write Without Response	None	Encryption required
Broadcast Receive State	M	Read, Notify	None	Encryption required

Table 3.1: BASS characteristics

There shall be only one Broadcast Audio Scan Control Point characteristic.

There shall be one or more Broadcast Receive State characteristics.

Servers should expose a number of Broadcast Receive State characteristics at least equal to the number of BIGs that the server can simultaneously maintain synchronization to.

3.1 Broadcast Audio Scan Control Point

When written by a client, the Broadcast Audio Scan Control Point characteristic is defined as an 8-bit enumerated value, known as the opcode, followed by zero or more parameter octets.

Broadcast Audio Scan Control Point opcodes are defined in [Table 3.2](#).

Opcode	Operation	Requirement	Section Reference	Description
0x00	Remote Scan Stopped	C.1	Section 3.1.1.2	Informs the server that the client is not scanning for Broadcast Sources on behalf of the server.
0x01	Remote Scan Started	C.1	Section 3.1.1.3	Informs the server that the client is scanning for Broadcast Sources on behalf of the server.
0x02	Add Source	C.1	Section 3.1.1.4	Requests the server to add information including Metadata for a Broadcast Source, and requests the server to synchronize to a PA and/or BIS transmitted by the Broadcast Source.
0x03	Modify Source	C.1	Section 3.1.1.5	Requests the server to update Metadata, to synchronize to, or to stop synchronizing to a PA and/or BIS transmitted by the Broadcast Source identified by the Source_ID.

Opcode	Operation	Requirement	Section Reference	Description
0x04	Set Broadcast_Code	M	Section 3.1.1.6	Provides the server with the Broadcast_Code to decrypt a BIS transmitted by a Broadcast Source identified by the Source_ID.
0x05	Remove Source	C.1	Section 3.1.1.7	Requests the server to remove all information for a Broadcast Source identified by the Source_ID.
0x06-0xFF	RFU	—	—	—

Table 3.2: Broadcast Audio Scan Control Point opcodes

C.1: If any one of these operations is supported, all of these operations shall be supported.

3.1.1 Characteristic behavior

When the Broadcast Audio Scan Control Point characteristic is written by a client, the first octet of that value shall be interpreted as the opcode. If the server accepts the operation defined by that opcode, the server shall perform the behavior defined for that operation.

3.1.1.1 Error Handling

If the server detects the total length of the opcode and parameter values written using the GATT Write Without Response sub-procedure by a client for any operation defined in [Section 3.1.1.2](#) to [Section 3.1.1.7](#) to be less than or greater than the total length of the opcode and all fixed length parameter values plus the length of any variable length parameter values for that operation, the server shall ignore the operation.

If the server detects the total length of the opcode and parameter values, written by a client by using the GATT Write Characteristic Value sub-procedure, or any operation defined in [Section 3.1.1.2](#) to [Section 3.1.1.7](#) to be less than or greater than the total length of the opcode and all fixed length parameter values plus the length of any variable length parameter values for that operation, the server shall respond with an ATT Error Response and shall set the Error Code parameter to Write Request Rejected as defined in [\[4\]](#).

If the server detects that an Opcode parameter value, written by a client by using the GATT Write Characteristic Value sub-procedure, is not supported, the server shall respond with an ATT Error Response and shall set the Error Code parameter to Opcode Not Supported as defined in [Table 1.2](#).

If the server detects that a Source_ID parameter value written by a client by using the GATT Write Characteristic Value sub-procedure does not match a Source_ID parameter value exposed in any Broadcast Receive State characteristic value, the server shall respond with an ATT Error Response and shall set the Error Code parameter to Invalid Source_ID as defined in [Table 1.2](#).

If the server detects that a Source_ID parameter value written by a client by using the GATT Write Without Response sub-procedure does not match a Source_ID parameter value exposed in any Broadcast Receive State characteristic value, the server shall ignore the operation.

If the server detects that a BIS_Sync parameter value written by a client is not 0xFFFFFFFF for a subgroup, and if the server detects that a BIS_index value written by a client is set to a value of 0b1 in more than one subgroup, the server shall ignore the operation.

3.1.1.2 Remote Scan Stopped operation

The Remote Scan Stopped operation is used to inform the server that the client is not scanning on behalf of the server. The server may modify its own scanning behavior as a result of a client writing the Remote Scan Stopped operation.

The Remote Scan Stopped operation has the format defined in [Table 3.3](#).

Opcode	Size (Octets)	Description
0x00	1	0x00 = Remote Scan Stopped operation
Parameter	Size	Description
None	0	Shall not contain any data

Table 3.3: Format of Remote Scan Stopped operation

3.1.1.3 Remote Scan Started operation

The Remote Scan Started operation is used to inform the server that the client is scanning on behalf of the server. The server may modify its scanning behavior as a result of a client writing the Remote Scan Started operation.

The Remote Scan Started operation has the format defined in [Table 3.4](#).

Opcode	Size (Octets)	Description
1	1	0x01 = Remote Scan Started operation
Parameter	Size	Description
None	0	Shall not contain any data

Table 3.4: Format of Remote Scan Started operation

3.1.1.4 Add Source operation

The Add Source operation is used to provide the server with information regarding a Broadcast Source.

The Add Source operation has the format defined in [Table 3.5](#).

Opcode	Size (Octets)	Description
0x02	1	0x02 = Add Source operation
Parameter	Size (Octets)	Description

Advertiser_Ad- dress_Type	1	Advertiser_Address_Type for the Broadcast Source 0x00 = Public Device Address or Public Identity Address 0x01 = Random Device Address or Random (static) Identity Address All other values: RFU
Advertiser_- Address	6	Advertiser_Address for the Broadcast Source
Advertising_SID	1	Advertising_SID subfield of the ADI field of the AUX_ADV_IND PDU or the LL_PERIODIC_SYNC_IND containing the SyncInfo that points to the PA transmitted by the Broadcast Source. Range: 0x00-0x0F All other values: RFU
Broadcast_ID	3	Broadcast_ID of the Broadcast Source
PA_Sync	1	0x00: Do not synchronize to PA 0x01: Synchronize to PA – PAST available 0x02: Synchronize to PA – PAST not available All other values: RFU
PA_Interval	2	SyncInfo field Interval parameter value 0xFFFF: PA_Interval unknown
Num_Subgroups	1	Number of subgroups
BIS_Sync[i]	4	BIS_Sync parameter for the [i th] subgroup in the BIG 4-octet bitfield. Bit 0-30 = BIS_index[1-31] 0x00000000: 0b0 = Do not synchronize to BIS_index[x] 0xxxxxxx: 0b1 = Synchronize to BIS_index[x] 0xFFFFFFFF: No preference Shall not exist if Num_Subgroups = 0
Metadata_- Length[i]	1	Length of the Metadata parameter value for the [i th] subgroup in the BIG Shall not exist if Num_Subgroups = 0
Metadata[i]	Varies	LTV-formatted Metadata for the [i th] subgroup in the BIG Shall exist only if the Metadata_Length parameter value is ≠ 0x00

Table 3.5: Format of Add Source operation

If the server accepts the Add Source operation, the server shall select an empty Broadcast Receive State characteristic to update, or if the server has no empty Broadcast Receive State characteristics, the server shall first delete all fields in a selected Broadcast Receive State characteristic, and:

- The server shall write a unique Source_ID value to the Source_ID field of the selected Broadcast Receive State characteristic.
- The server shall write the value of the Advertiser_Address_Type parameter [2] written by the client to the Source_Address_Type field of the selected Broadcast Receive State characteristic.

- The server shall write the value of the Advertiser_Address parameter [2] written by the client to the Source_Address field of the selected Broadcast Receive State characteristic.
- The server shall write the value of the Advertising_SID parameter [2] written by the client to the Source_Adv_SID field of the selected Broadcast Receive State characteristic.
- The server shall write the value of the Broadcast_ID parameter [3] written by the client to the Broadcast_ID field of the selected Broadcast Receive State characteristic.
- If the PA_Sync parameter value written by the client is set to a value of 0x00 (Do not synchronize to PA), the server shall write a value of 0x00 (Not synchronized to PA) to the PA_Sync_State field of the selected Broadcast Receive State characteristic and the server shall not attempt to synchronize to the PA.
- If the PA_Sync parameter value written by the client is set to a value of 0x01 (Synchronize to PA – PAST available), or 0x02 (Synchronize to PA – PAST not available), the server shall either write a value of 0x01 (SyncInfo Request) to the PA_Sync_State field of the selected Broadcast Receive State characteristic to request a client to transfer SyncInfo data to the server or the server shall attempt to synchronize with the PA by using the Periodic Advertising Synchronization Establishment procedure defined in Volume 3, Part C, Section 9.5.3 in [2].
 - If the Advertiser_Address_Type value written by the client is 0x01, the Advertiser_Address value written by the client might not match the AdvA field in ADV_EXT_IND PDUs transmitted by the Broadcast Source (for example, because the client's Bluetooth Controller is performing Private device address resolution as defined in Vol 6, Part B, Section 1.3.2.3 in [2]). If the server attempts to synchronize with the PA, the server may determine the Advertiser_Address to be used in the Periodic Advertising Synchronization Establishment procedure by:
 - Comparing the Adv_SID written by the client to the Adv_SID subfield of the ADI field of ADV_EXT_IND PDUs transmitted by the Broadcast Source.
 - Comparing the Broadcast_ID written by the client to the Broadcast_ID in the AdvData field of AUX_ADV_IND PDUs transmitted by the Broadcast Source.
- The server shall only write a value of 0x01 (SyncInfo Request) to the PA_Sync_State field of the selected Broadcast Receive State characteristic if the server supports the PAST procedure defined in Volume 3, Part C, Section 9.5.4 in [2].
- If the server has written a value of 0x01 (SyncInfo Request) to the PA_Sync_State field of the selected Broadcast Receive State characteristic and the server does not receive SyncInfo data from a client within a time period defined by the implementation or by a higher-layer specification, the server shall write a value of 0x04 (No PAST) to the PA_Sync_State field of the Broadcast Receive State characteristic and the server shall not attempt to synchronize to the PA.
- If the server has received SyncInfo data from a client, the server shall attempt to synchronize to the PA by using the Periodic Advertising Synchronization Establishment procedure defined in Volume 3, Part C, Section 9.5.3 in [2].
- If the server synchronizes to the PA, the server shall write a value of 0x02 (Synchronized to PA) to the PA_Sync_State field of the selected Broadcast Receive State characteristic. If the server fails to synchronize with the PA, the server shall write a value of 0x03 (Failed to synchronize to PA) to the PA_Sync_State field of the selected Broadcast Receive State characteristic.
- If the server has synchronized to the PA and the server has detected that the BIS is encrypted, and if the server does not have an encryption key to decrypt the BIS, the server shall write a value of 0x01 (Broadcast_Code required) to the BIG_Encryption field of the selected Broadcast Receive State characteristic to request a client to provide a Broadcast_Code.
- The server shall write the value of the Num_Subgroups parameter to the Num_Subgroups field of the Broadcast Receive State characteristic.

- For each subgroup, if the server has synchronized to the PA, and if the BIS_Sync parameter value written by the client is not 0xFFFFFFFF, the server shall attempt to synchronize to each BIS whose BIS_index value is set to 0b1 only if the BIS_index value is not set to 0b1 in any other subgroup. However, if the BIS_Sync parameter value written by the client is set to 0xFFFFFFFF (No preference), the server may attempt to synchronize to any BIS in the BIG. To synchronize to a BIS, the server shall use the Broadcast Isochronous Synchronization Establishment procedure defined in Volume 3, Part C, Section 9.6.3 in [2].
- For each subgroup, if the server has synchronized to a BIS, the server shall set the BIS_index value of that BIS to 0b1 for the subgroup containing that BIS in the BIS_Sync field of the selected Broadcast Receive State characteristic.
- If the server has synchronized to a BIS and if the server has detected that the BIS is encrypted, and if the server does not have an encryption key to decrypt the BIS, the server shall write a value of 0x01 (Broadcast_Code required) to the BIG_Encryption field of the selected Broadcast Receive State characteristic to request a client to provide a Broadcast_Code.
- If the BIS is encrypted and the server has the correct encryption key to decrypt the BIS, the server shall write a value of 0x02 (Decrypting) to the BIG_Encryption field of the selected Broadcast Receive State characteristic.
- If the BIS is encrypted and the server has detected that a Broadcast_Code parameter value written by a client is not the correct encryption key to decrypt the BIS, the server shall write a value of 0x03 (Bad_Code) to the BIG_Encryption field of the selected Broadcast Receive State characteristic and the server shall write the value 0xFFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF to the Bad_Code field of the selected Broadcast Receive State characteristic.
- If the BIS is not encrypted, the server shall write a value of 0x00 (Not encrypted) to the BIG_Encryption field of the selected Broadcast Receive State characteristic.
- For each BIS whose BIS_index value is set to 0b0 in the BIS_Sync parameter value written by the client for every subgroup, the server shall not attempt to synchronize to that BIS and shall set the BIS_index value of that BIS to 0b0 in the BIS_Sync_State field of the selected Broadcast Receive State characteristic.
- For each subgroup, for each BIS the server attempts to synchronize to and fails, the server shall set the BIS_index value of that BIS to 0b0 in the BIS_Sync_State field of the Broadcast Receive State characteristic. However, if the server fails to synchronize to the BIG, the server shall write a value of 0xFFFFFFFF (Failed to synchronize to BIG) to the BIS_Sync_State field of the selected Broadcast Receive State characteristic.
- The server shall not attempt to synchronize to a BIS if the server has not synchronized to the PA.
- If the server has synchronized to the PA and then loses synchronization to the PA, the server shall write a value of 0x00 (Not synchronized to PA) to the PA_Sync_State field of the selected Broadcast Receive State characteristic.
- For each subgroup, if the server has synchronized to a BIS and then loses synchronization to that BIS, the server shall set the BIS_index value of that BIS to 0b0 in the BIS_Sync_State field of the selected Broadcast Receive State characteristic.
- For each subgroup, if the Metadata_Length parameter value written by the client is nonzero, the Metadata parameter value written by the client may be written to the Metadata field of the selected Broadcast Receive State characteristic.

If the server does not accept the Add Source operation, the server shall not modify the value of any Broadcast Receive State characteristic with the values written by the client in the Add Source operation.

3.1.1.5 Modify Source operation

The Modify Source operation is used to request the server to add or update Metadata for the Broadcast Source, and/or to request the server to synchronize to, or to stop synchronization to, a PA and/or a BIS.

The Modify Source operation has the format defined in Table 3.6.

Opcode	Size (Octets)	Description
0x03	1	0x03 = Modify Source operation
Parameter	Size (Octets)	Description
Source_ID	1	Source_ID assigned by the server to a Broadcast Receive State characteristic
PA_Sync	1	0x00 = Do not synchronize to PA 0x01 = Synchronize to PA – PAST available 0x02 = Synchronize to PA – PAST not available All other values: RFU
PA_Interval	2	SyncInfo field Interval parameter value 0xFFFF: PA_Interval unknown
Num_Subgroups	1	Number of subgroups
BIS_Sync[i]	4	BIS_Sync parameter for the [i th] subgroup in the BIG 4-octet bitfield. Bit 0-30 = BIS_index[1-31] 0x00000000: 0b0 = Do not synchronize to BIS_index[x] 0xxxxxxx: 0b1 = Synchronize to BIS_index[x] 0xFFFFFFFF: No preference Shall not exist if Num_Subgroups = 0
Metadata_Length[i]	1	Length of the Metadata parameter value for the [i th] subgroup in the BIG Shall not exist if Num_Subgroups = 0
Metadata[i]	Varies	LTV-formatted Metadata for the [i th] subgroup in the BIG for the Broadcast Source identified by the Source_ID Shall exist only if the Metadata_Length parameter value is ≠ 0x00

Table 3.6: Format of Modify Source operation

If the server accepts the Modify Source operation, the server shall select the Broadcast Receive State characteristic containing a Source_ID value that matches the Source_ID written by the client during the Modify Source operation, and:

- If the PA_Sync parameter value written by the client is set to a value of 0x00 (Do not synchronize to PA) and the server is not synchronized to the PA, the server shall write a value of 0x00 (Not synchronized to PA) to the PA_Sync_State field of the Broadcast Receive State characteristic and the server shall not attempt to synchronize to the PA.

- If the PA_Sync parameter value written by the client is set to a value of 0x00 (Do not synchronize to PA) and the server is synchronized to the PA, the server shall stop synchronization with the PA and shall write a value of 0x00 (Not synchronized to PA) to the PA_Sync_State field of the Broadcast Receive State characteristic .
- If the PA_Sync parameter value written by the client is set to a value of 0x01 (Synchronize to PA – PAST available), or 0x02 (Synchronize to PA – PAST not available), the server shall either write a value of 0x01 (SyncInfo Request) to the PA_Sync_State field of the Broadcast Receive State characteristic if the server requires a client to transfer SyncInfo data to the server, or the server shall attempt to synchronize with the PA by using the Periodic Advertising Synchronization Establishment procedure defined in Volume 3, Part C, Section 9.5.3 in [2].
 - If the Source_Address_Type field of the Broadcast Receive State characteristic is 0x01, the Source_Address field of the Broadcast Receive State characteristic might not match the AdvA field in ADV_EXT_IND PDUs transmitted by the Broadcast Source. If the server attempts to synchronize with the PA, the server may determine the Advertiser_Address to be used in the Periodic Advertising Synchronization Establishment procedure by:
 - Comparing the Adv_SID field of the Broadcast Receive State characteristic to the Adv_SID subfield of the ADI field of ADV_EXT_IND PDUs transmitted by the Broadcast Source.
 - Comparing the Broadcast_ID field of the Broadcast Receive State characteristic to the AdvData field of AUX_ADV_IND PDUs transmitted by the Broadcast Source.
- The server shall only write a value of 0x01 (SyncInfo Request) to the PA_Sync_State field of the Broadcast Receive State characteristic if the server supports the PAST procedure defined in Volume 3, Part C, Section 9.5.4 in [2].
- If the server has written a value of 0x01 (SyncInfo Request) to the PA_Sync_State field of the Broadcast Receive State characteristic and the server does not receive SyncInfo data from a client within a time period defined by the implementation or by a higher-layer specification, the server shall write a value of 0x04 (No PAST) to the PA_Sync_State field of the Broadcast Receive State characteristic and the server shall not attempt to synchronize to the PA.
- If the server has received SyncInfo data from a client, the server shall attempt to synchronize to the PA by using the Periodic Advertising Synchronization Establishment procedure defined in Volume 3, Part C, Section 9.5.3 in [2].
- If the server has synchronized to the PA, the server shall write a value of 0x02 (Synchronized to PA) to the PA_Sync_State field of the Broadcast Receive State characteristic; otherwise, the server shall write a value of 0x03 (Failed to synchronize to PA) to the PA_Sync_State field of the Broadcast Receive State characteristic.
- If the server has synchronized to the PA, and the server has detected that the BIS is encrypted, and if the server does not have the correct encryption key to decrypt the BIS, the server shall write a value of 0x01 (Broadcast_Code required) to the BIG_Encryption field of the Broadcast Receive State characteristic to request a client to provide a Broadcast_Code.
- The server shall write the value of the Num_Subgroups parameter to the Num_Subgroups field of the Broadcast Receive State characteristic.
- For each subgroup, if the server has synchronized to the PA, and if the BIS_Sync parameter value written by the client is not 0xFFFFFFFF, the server shall attempt to synchronize to each BIS whose BIS_index value is set to 0b1 only if the BIS_index value is not set to 0b1 in any other subgroup in the BIS_Sync parameter value written by the client. However, if the BIS_Sync parameter value written by the client is set to 0xFFFFFFFF (No preference), the server may attempt to synchronize to any BIS in the BIG. To synchronize to a BIS, the server shall use the Broadcast Isochronous Synchronization Establishment procedure defined in Volume 3, Part C, Section 9.6.3 in [2].

- For each subgroup, if the server has synchronized to a BIS, the server shall set the BIS_index value of that BIS to 0b1 for the subgroup containing that BIS in the BIS_Sync_State field of the Broadcast Receive State characteristic value.
- If the server has synchronized to a BIS and the server has detected that the BIS is encrypted, and if the server does not have an encryption key to decrypt the BIS, the server shall write a value of 0x01 (Broadcast_Code required) to the BIG_Encryption field of the Broadcast Receive State characteristic to request a client to provide a Broadcast_Code.
- If the BIS is encrypted and the server has the correct encryption key to decrypt the BIS, the server shall write a value of 0x02 (Decrypting) to the BIG_Encryption field of the Broadcast Receive State characteristic.
- If the BIS is encrypted and the server has detected that a Broadcast_Code parameter value written by a client is not the correct encryption key to decrypt the BIS, the server shall write a value of 0x03 (Bad_Code) to the BIG_Encryption field of the Broadcast Receive State characteristic and the server shall write the value 0xFFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF to the Bad_Code field of the selected Broadcast Receive State characteristic.
- If the BIS is not encrypted, the server shall write a value of 0x00 (Not encrypted) to the BIG_Encryption field of the Broadcast Receive State characteristic.
- For each BIS whose BIS_index value is set to 0b0 in the BIS_Sync field written by the client for every subgroup, the server shall not attempt to synchronize to that BIS, or if the server is synchronized to that BIS, the server shall stop synchronization with that BIS and shall set the BIS_index value of that BIS to 0b0 in the BIS_Sync_State field of the Broadcast Receive State characteristic.
- For each subgroup, for each BIS the server attempts to synchronize to and fails, the server shall set the BIS_index value of that BIS to 0b0 in the BIS_Sync_State field of the Broadcast Receive State characteristic. However, if the server fails to synchronize to the BIG, the server shall write a value of 0xFFFFFFFF (Failed to synchronize to BIG) to the BIS_Sync_State field of the Broadcast Receive State characteristic.
- The server shall not attempt to synchronize to a BIS if the server has not synchronized to the PA.
- If the server has synchronized to the PA and then loses synchronization to the PA, the server shall write a value of 0x00 (Not synchronized to PA) to the PA_Sync_State field of the Broadcast Receive State characteristic.
- For each subgroup, if the server has synchronized to a BIS and then loses synchronization to that BIS, the server shall set the BIS_index value of that BIS to 0b0 in the BIS_Sync_State field of the Broadcast Receive State characteristic.
- For each subgroup, if the Metadata_Length parameter value written by the client is nonzero, the Metadata parameter value written by the client may be written by the server to the Metadata field of the Broadcast Receive State characteristic.

If the server does not accept the Modify Source operation, the server shall not modify the value of the Broadcast Receive State characteristic that contains a Source_ID value that matches the Source_ID value written by the client in the Modify Source operation.

3.1.1.6 Set Broadcast_Code operation

The Set Broadcast_Code operation is used by a client to provide a Broadcast_Code to the server to enable the server to decrypt an encrypted BIS.

The Set Broadcast_Code operation has the format defined in [Table 3.7](#).



Opcode	Size (Octets)	Description
0x04	1	0x04 = Set Broadcast_Code operation
Parameter	Size (Octets)	Description
Source_ID	1	Source_ID assigned by the server to a Broadcast Receive State characteristic
Broadcast_Code	16	Broadcast_Code for the Source_ID assigned to a Broadcast Receive State characteristic

Table 3.7: Format of Set Broadcast_Code operation

The server shall use the provided Broadcast_Code value to attempt to decrypt an encrypted BIS.

3.1.1.7 Remove Source operation

The Remove Source operation is used by a client to request the server to remove information for a Broadcast Source identified by the Source_ID in a Broadcast Receive State characteristic.

The Remove Source operation has the format defined in Table 3.8.

Opcode	Size (Octets)	Description
0x05	1	0x05 = Remove Source operation
Parameter	Size (Octets)	Description
Source_ID	1	Source_ID assigned by the server to a Broadcast Receive State characteristic

Table 3.8: Format of Remove Source operation

If the server accepts the Remove Source operation, the server shall delete all fields of the Broadcast Receive State characteristic containing the Source_ID value that matches the Source_ID value written by the client in the Remove Source operation.

The server shall not accept a Remove Source operation for a Source_ID value that matches the Source_ID written by the client in the Remove Source operation if the server is synchronized to the PA and/or any BIS as defined by the values of the PA_Sync_State and BIS_Sync_State fields in the Broadcast Receive State characteristic containing that Source_ID value.

If the server does not accept the Remove Source operation, the server shall not modify the value of the Broadcast Receive State characteristic that contains a Source_ID value that matches the Source_ID value written by the client in the Remove Source operation.

3.2 Broadcast Receive State

The Broadcast Receive State characteristic is used by the server to expose information about a Broadcast Source, representing the current synchronization state of the server to a PA and/or a BIG

containing one or more subgroups containing one or more BISes transmitted by that Broadcast Source. The Broadcast Receive State characteristic is also used to inform clients whether the server has detected that the BIS is encrypted, whether the server requires a Broadcast_Code, and whether the server is decrypting the BIS.

When the Broadcast Receive State characteristic value is not empty (zero length), the Broadcast Receive State characteristic has the format defined in [Table 3.9](#).

Field	Size (Octets)	Description
Source_ID	1	Assigned by the server Shall be unique for each instance of the Broadcast Receive State characteristic exposed by the server Range: 0x00-0xFF
Source_Address_Type	1	0x00 = Public Device Address or Public Identity Address 0x01 = Random Device Address or Random (static) Identity Address All other values: RFU
Source_Address	6	Public Device Address, Random Device Address, Public Identity Address or Random (static) Identity Address of the Broadcast Source.
Source_Adv_SID	1	Advertising_SID subfield of the ADI field of the AUX_ADV_IND PDU or the LL_PERIODIC_SYNC_IND containing the SyncInfo that points to the PA transmitted by the Broadcast Source.
Broadcast_ID	3	Broadcast_ID of the Broadcast Source
PA_Sync_State	1	0x00: Not synchronized to PA 0x01: SyncInfo Request 0x02: Synchronized to PA 0x03: Failed to synchronize to PA 0x04: No PAST All other values: RFU
BIG_Encryption	1	0x00: Not encrypted 0x01: Broadcast_Code required 0x02: Decrypting 0x03: Bad_Code (incorrect encryption key) All other values: RFU
Bad_Code	Varies	If BIG_Encryption field value = 0x00, 0x01, or 0x02: empty (zero length) If BIG_Encryption field value = 0x03 (Bad_Code), Bad_Code shall be set to the value 0xFFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
Num_Subgroups	1	Number of subgroups

Field	Size (Octets)	Description
BIS_Sync_State[i]	4	BIS_Sync_State for the [i th] subgroup 4-octet bitfield. Bit 0-30 = BIS_index[1-31] 0x00000000: 0b0 = Not synchronized to BIS_index[x] 0xxxxxxx: 0b1 = Synchronized to BIS_index[x] 0xFFFFFFFF: Failed to sync to BIG Shall not exist if Num_Subgroups = 0
Metadata_Length[i]	1	Length of the Metadata field for the [i th] subgroup Shall not exist if Num_Subgroups = 0
Metadata[i]	Varies	LTV-formatted Metadata for the [i th] subgroup Shall exist only if the Metadata_Length parameter value is ≠ 0x00

Table 3.9: Broadcast Receive State characteristic format

3.2.1 Characteristic behavior

If the server has not written a Source_ID value to the Broadcast Receive State characteristic, the Broadcast Receive State characteristic value shall be empty (zero length).

If the Broadcast Receive State characteristic is not empty when read by a client, the Broadcast Receive State characteristic returns the current Source_ID, Source_Address_Type, Source_Address, Source_Adv_SID, and Broadcast_ID values used to identify a Broadcast Source, and the current PA_Sync_State, BIG_Encryption, Num_Subgroups, BIS_Sync_State, Metadata_Length, and Metadata values.

The server may modify the Broadcast Receive State characteristic value autonomously or when accepting an operation written by the client using the Broadcast Audio Scan Control Point characteristic.

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

If the Broadcast Receive State characteristic value changes when in a connection to a client, the server shall notify the new characteristic value to the client.

If the Broadcast Receive State characteristic value is not empty (zero length), and if a bonded client reconnects to the server, the server shall notify the current Broadcast Receive State characteristic value to that bonded client.

3.2.1.1 Source_ID field

If the server has autonomously synchronized to a PA, the server shall write a unique value to the Source_ID field.

If the server has accepted an Add Source operation written by a client to the Broadcast Audio Scan Control Point characteristic, the server shall follow the behavior defined in [Section 3.1.1.4](#) to write the value of the Source_ID field.



3.2.1.2 Source_Address_Type field

If the server has autonomously synchronized to a PA, the server shall write the value of the Broadcast Source's Advertiser_Address_Type parameter to the Source_Address_Type field.

If the server has accepted an Add Source operation written by a client to the Broadcast Audio Scan Control Point characteristic, the server shall follow the behavior defined in [Section 3.1.1.4](#) to write the value of the Source_Address_Type field.

3.2.1.3 Source_Address field

If the server has autonomously synchronized to a PA, the server shall write the value of the Broadcast Source's Advertiser_Address parameter to the Source_Address field.

If the server is aware the Advertiser_Address for a Broadcast Source has changed, the server shall write the updated Advertiser_Address to the Source_Address field.

If the server has accepted an Add Source operation written by a client to the Broadcast Audio Scan Control Point characteristic, the server shall follow the behavior defined in [Section 3.1.1.4](#) to write the value of the Source_Address field.

If the server receives a transfer of SyncInfo data from a client in an LL_PERIODIC_SYNC_IND PDU as defined in Vol 6, Part B, Section 2.4.2.27 in [\[2\]](#), and if the server is not using the Host Controller Interface (HCI), and if the value received in octet 0 of the ID parameter is 0x02 or 0x03, the server shall write the received AdvA parameter value to the Source_Address field of the Broadcast Receive State characteristic with the Source_ID field that matches the Source_ID value received in octet 1 of the ID parameter.

If the server receives an HCI_LE_Periodic_Advertising_Sync_Transfer_Received_Event as defined in Vol 4, Part E, Section 7.7.65.24 in [\[2\]](#), and if the value received in octet 0 of the Service_Data parameter is 0x02 or 0x03, the server shall write the received Advertiser_Address parameter value to the Source_Address field of the Broadcast Receive State characteristic with the Source_ID field that matches the Source_ID value received in octet 1 of the Service_Data parameter.

3.2.1.4 Source_Adv_SID field

If the server has autonomously synchronized to a PA, the server shall write the value of the Advertising_SID subfield of the ADI field of the AUX_ADV_IND PDU [\[2\]](#) used to transport the PA to the Source_Adv_SID field.

If the server has accepted an Add Source operation written by a client to the Broadcast Audio Scan Control Point characteristic, the server shall follow the behavior defined in [Section 3.1.1.4](#) to write the value of the Source_Adv_SID field.

3.2.1.5 Broadcast_ID field

If the server has autonomously synchronized to a PA, the server shall write the value of the Broadcast_ID in the AdvData field of the AUX_ADV_IND PDU to the Broadcast_ID field.

If the server has accepted an Add Source operation written by a client to the Broadcast Audio Scan Control Point characteristic, the server shall follow the behavior defined in [Section 3.1.1.4](#) to write the value of the Broadcast_ID field.

3.2.1.6 PA_Sync_State field

The PA_Sync_State field is used to expose the current synchronization state of the server with respect to a PA.

If the server has accepted an Add Source operation written by the client, the server shall follow the behavior defined in [Section 3.1.1.4](#) to write the value of the PA_Sync_State field.

If the server has accepted a Modify Source operation written by the client, the server shall follow the behavior defined in [Section 3.1.1.5](#) to write the value of the PA_Sync_State field.

The server may write a value of 0x01 (SyncInfo Request) to the PA_Sync_State field at any time to request a client to provide synchronization information for a PA.

If the server has autonomously synchronized to a PA, the server shall write a value of 0x02 (Synchronized to PA) to the PA_Sync_State field of the Broadcast Receive State characteristic.

If the server has autonomously synchronized to a PA and detected, by checking the length of the BIGInfo [2], that a BIS is encrypted, the server shall follow the behavior in [Section 3.2.1.7](#) to request a Broadcast_Code.

If the server has autonomously attempted and failed to synchronize to a PA, the server shall write a value of 0x00 (Not synchronized to PA) to the PA_Sync_State field of the Broadcast Receive State characteristic.

If the server has autonomously synchronized to a PA and then lost or stopped synchronization with the PA, the server shall write a value of 0x00 (Not synchronized to PA) to the PA_Sync_State field of the Broadcast Receive State characteristic.

3.2.1.7 BIG_Encryption field

The BIG_Encryption field is used to expose the encryption state of one or more BISes that the server is synchronized to.

If the server has accepted an Add Source operation written by the client, the server shall follow the behavior defined in [Section 3.1.1.4](#) if the server has accepted an Add Source operation written by the client to write the value of BIG_Encryption field.

If the server has accepted a Modify Source operation written by the client, the server shall follow the behavior defined in [Section 3.1.1.5](#) to write the value of the BIG_Encryption field.

If the server has detected that a BIS is encrypted, or if the server has autonomously synchronized to a BIS that is encrypted, and the server does not have an encryption key to decrypt the BIS, the server shall write a value of 0x01 (Broadcast_Code required) to the BIG_Encryption field of the Broadcast Receive State characteristic to inform clients that the server requires a client to write the Set Broadcast_Code operation to the Broadcast Audio Scan Control Point characteristic with the correct encryption key to decrypt the BIS.

If the server has autonomously synchronized to a BIS that is encrypted, and the server has the correct encryption key to decrypt the BIS, the server shall write a value of 0x02 (Decrypting) to the BIG_Encryption field of the Broadcast Receive State characteristic.

If the server has autonomously synchronized to a BIS that is encrypted, and the server has detected that a received encryption key is not the correct encryption key to decrypt the BIS (the server can receive encryption keys via writes to the Broadcast Audio Scan Control Point or by other means), the server

shall write a value of 0x03 (Bad_Code) to the BIG_Encryption field of the Broadcast Receive State characteristic and the server shall write the value 0xFFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF to the Bad_Code field of the Broadcast Receive State characteristic.

If the server has autonomously synchronized to a BIS that is not encrypted, the server shall write a value of 0x00 (Not encrypted) to the BIG_Encryption field of the Broadcast Receive State characteristic.

If the server has not determined whether the BIS is encrypted (for example, because the server has not yet received the BIGInfo from the PA), the server shall write a value 0x00 (Not encrypted) to the BIG_Encryption field of the Broadcast Receive State characteristic.

3.2.1.8 Num_Subgroups field

The Num_Subgroups field is used to expose the number of subgroups present in the Broadcast Audio Source Endpoint (BASE) structure, defined in Section 3.7.2.2 in [3], used to describe a BIG.

If the server has accepted an Add Source operation written by the client, the server shall follow the behavior defined in Section 3.1.1.4 to write the value of the Num_Subgroups field.

If the server has accepted a Modify Source operation written by the client, the server shall follow the behavior defined in Section 3.1.1.5 to write the value of the Num_Subgroups field.

If the server has autonomously synchronized to a PA, the server shall set the Num_Subgroups field to the value of the Num_Subgroups parameter present in the BASE structure of that PA.

3.2.1.9 BIS_Sync_State field

The BIS_Sync_State field is used to expose the synchronization state of the server with respect to one or more BISes for one or more subgroups in a BIG.

If the server has accepted an Add Source operation written by the client, the server shall follow the behavior defined in Section 3.1.1.4 to write the value of the BIS_Sync_State field.

If the server has accepted a Modify Source operation written by the client, the server shall follow the behavior defined in Section 3.1.1.5 to write the value of the BIS_Sync_State field.

If the server has autonomously synchronized to a BIS, the server shall set the BIS_index value of that BIS to 0b1 for the subgroup containing that BIS in the BIS_Sync_State field of the Broadcast Receive State characteristic.

If the server has autonomously synchronized to a BIS and then lost or stopped synchronization with that BIS, the server shall set the BIS_index value of that BIS to 0b0 for the subgroup containing that BIS in the BIS_Sync_State field of the Broadcast Receive State characteristic.

3.2.1.10 Metadata_Length field

If the server has autonomously synchronized to the PA, the server may write the length of any Metadata parameters for each subgroup to the Metadata_Length field.

If the server has accepted an Add Source operation written by the client, the server shall follow the behavior defined in Section 3.1.1.4 to write the value of the Metadata_Length field.

If the server has accepted a Modify Source operation written by the client, the server shall follow the behavior defined in Section 3.1.1.5 to write the value of the Metadata_Length field.

3.2.1.11 Metadata field

If the server has autonomously synchronized to the PA, the server may write Metadata values for each subgroup to the Metadata field of the Broadcast Receive State characteristic, including any Metadata values not present in the BASE that the server wishes to include.

The server shall follow the behavior defined in [Section 3.1.1.4](#) if the server has accepted an Add Source operation written by the client to write the value of Metadata field.

The server shall follow the behavior defined in [Section 3.1.1.5](#) if the server has accepted a Modify Source operation written by the client to write the value of the Metadata field.

4 SDP interoperability

When exposing this service over Basic Rate/Enhanced Data Rate (BR/EDR), include the following SDP Record attributes and values.

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

Item	Definition	Type	Value	Requirement
Service Class ID List	–	–	–	M
Service Class #0	–	UUID	«Broadcast Audio Scan»	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
BASE	Broadcast Audio Source Endpoint
BASS	Broadcast Audio Scan Service
BIG	Broadcast Isochronous Group
BIS	Broadcast Isochronous Stream
BR/EDR	Basic Rate/Enhanced Data Rate
EATT	Enhanced ATT
GATT	Generic Attribute Profile
HCI	Host Controller Interface
L2CAP	Logical Link Control and Adaption Protocol
LL	Link Layer
LSO	least significant octet
PAST	Periodic Advertising Synchronization Transfer
PDU	Protocol Data Unit
PSM	Protocol/Service Multiplexer
RFU	Reserved for Future Use
UUID	universally unique identifier

Table 5.1: Acronyms and abbreviations

6 References

- [1] Bluetooth SIG Assigned Numbers, <https://www.bluetooth.com/specifications/assigned-numbers>
- [2] Bluetooth Core Specification (amended) Version 5.2 or later
- [3] Basic Audio Profile Specification, Version 1.0.1 or later
- [4] Bluetooth Core Specification Supplement, Version 12 or later