

# Volume Control Service

## **Bluetooth® Service Specification**

---

- **Revision:** v1.0
- **Revision Date:** 2020-12-15
- **Group Prepared By:** Generic Audio Working Group

### **Abstract:**

This specification describes the service that exposes a control interface and volume state on an audio device.



**Revision History**

Revision Number	Date	Comments
v1.0	2020-12-15	Adopted by the Bluetooth SIG Board of Directors.

**Contributors**

Name	Company
Michael Rougeux	Bose Corporation
Siegfried Lehmann	Apple, Inc.
Tim Reilly	Bose Corporation
Robin Heydon	Qualcomm, Inc.
Asbjørn Sæbø	Nordic Semiconductor ASA
Georg Dickmann	Sonova AG
Bjarne Klemmensen	Oticon A/S
HJ Lee	LG Electronics Inc.
Marcel Holtmann	Intel Corporation
Masahiko Seki	Sony Corporation
Søren Møllskov Larsen	Widex A/S
Daniel Sisolak	Bose Corporation
Oren Haggai	Intel Corporation
Frank Yerrace	Microsoft Corporation

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](http://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. Each option identified in the specification 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 © 2018–2020. All copyrights in the Bluetooth Specifications themselves are owned by Apple Inc., Ericsson AB, Intel Corporation, 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

<b>1</b>	<b>Introduction .....</b>	<b>6</b>
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	Language .....	7
1.8.1	Language conventions .....	7
1.8.2	Reserved for Future Use .....	7
1.8.3	Prohibited .....	8
1.8.4	Terminology .....	8
<b>2</b>	<b>Service .....</b>	<b>9</b>
2.1	Declaration .....	9
2.2	Included services .....	9
2.2.1	Topology .....	9
2.3	Overview .....	9
2.3.1	Volume State .....	9
2.3.1.1	Volume_Setting field .....	10
2.3.1.2	Mute field .....	10
2.3.1.3	Change_Counter field .....	10
2.3.2	Volume Flags .....	10
2.3.2.1	Volume_Setting_Persisted field .....	10
<b>3</b>	<b>Service characteristics .....</b>	<b>11</b>
3.1	Volume State .....	11
3.1.1	Volume_Setting field .....	11
3.1.2	Mute field .....	11
3.1.3	Change_Counter field .....	11
3.1.4	Volume State behavior .....	12
3.2	Volume Control Point .....	12
3.2.1	Volume Control Point procedure requirements .....	12
3.2.2	Volume Control Point behavior .....	12
3.2.2.1	Relative Volume Down procedure .....	13
3.2.2.2	Relative Volume Up procedure .....	13
3.2.2.3	Unmute/Relative Volume Down procedure .....	14
3.2.2.4	Unmute/Relative Volume Up procedure .....	14
3.2.2.5	Set Absolute Volume procedure .....	14
3.2.2.6	Unmute procedure .....	15
3.2.2.7	Mute procedure .....	15
3.3	Volume Flags .....	16
3.3.1.1	Volume Flags behavior .....	16
<b>4</b>	<b>SDP interoperability .....</b>	<b>17</b>



5 Acronyms and abbreviations..... 18

6 References..... 19



# 1 Introduction

This service enables a device to expose the controls and state of its audio volume.

## 1.1 Conformance

If conformance to this specification is claimed, all capabilities indicated as mandatory for this specification shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated.

## 1.2 Service dependencies

This service depends on the Volume Offset Control Service (VOCS) [3] and Audio Input Control Service (AICS) [4] if included by this service.

## 1.3 Bluetooth Core Specification release compatibility

This specification is compatible with any version of the Bluetooth Core Specification [1] that includes the Generic Attribute Profile (GATT).

## 1.4 GATT sub-procedure requirements

Requirements in this section represent a minimum set of server requirements. Other GATT sub-procedures may be used if supported by both the 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 over Unenhanced Attribute Protocol (ATT) bearers.

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

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

This service does not impose any additional GATT sub-procedure requirements beyond those required by all GATT servers over Enhanced ATT bearers.

## 1.5 Transport dependencies

This service uses GATT and therefore has no additional transport dependencies.

Notifications with GATT are considered unreliable when used with an Unenhanced ATT bearer.

An Enhanced ATT bearer can be used for reliability of Notifications and can be specified by a higher-layer profile.



## 1.6 Application error codes

This service defines the ATT Application Error codes shown in [Table 1.2](#).

Name	Error Code	Description
Invalid Change Counter	0x80	The Change_Counter operand value does not match the Change_Counter field value of the Volume State characteristic.
Opcode Not Supported	0x81	An invalid opcode has been used in a control point procedure.

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 the Bluetooth SIG Assigned Numbers [5].

## 1.8 Language

### 1.8.1 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.

### 1.8.2 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.

### 1.8.3 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.

### 1.8.4 Terminology

Table 1.3 defines terms that are needed to understand features used in this service.

Term	Definition
Unenhanced ATT bearer	An ATT bearer not 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 the Bluetooth Core Specification [2].
Enhanced ATT bearer	An ATT bearer using the Enhanced Credit Based Flow Control L2CAP channel mode introduced in Volume 3, Part A, Section 10.2 in [2].

Table 1.3: Terminology



## 2 Service

### 2.1 Declaration

There shall be no more than one instance of the Volume Control Service (VCS) on a device.

VCS is instantiated to expose the controls and state of a device that can control the volume of an audio output such as one or more speakers.

The Attribute Type service declaration shall be set to the «Primary Service» or «Secondary Service» universally unique identifier (UUID) and the Attribute Value service declaration shall be set to «Volume Control Service» as defined in the Bluetooth SIG Assigned Numbers [5].

### 2.2 Included services

VCS may include zero or more instances of VOCS [3] and zero or more instances of AICS [4].

#### 2.2.1 Topology

An example device topology using VCS and included instances of VOCS and AICS is shown in [Figure 2.1](#).

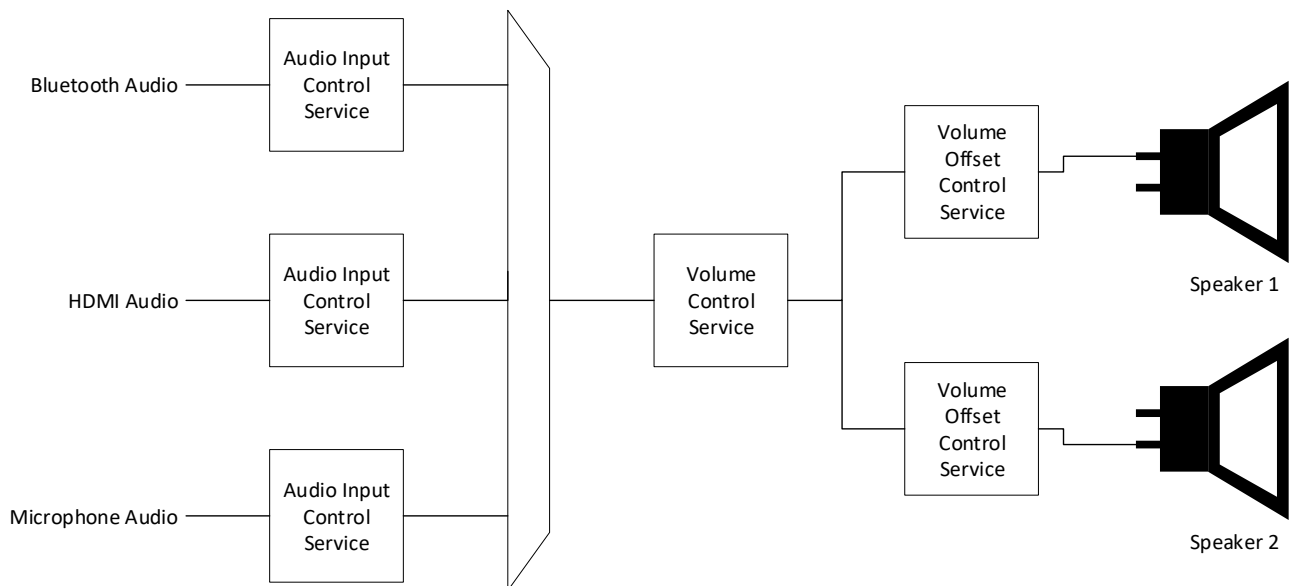


Figure 2.1: Example of VCS topology

### 2.3 Overview

This section provides an overview of the behavior of the characteristics that affect the server's audio volume and their usage.

#### 2.3.1 Volume State

The Volume State characteristic value is comprised of three fields: the Volume\_Setting, Mute, and Change\_Counter fields.

### 2.3.1.1 Volume\_Setting field

The Volume\_Setting field is a unitless value such that a step increase in its value should result in a step increase of the audio output volume and a step decrease in its value should result in a step decrease of the audio output volume (a step size is implementation specific).

A value of 0 of the Volume\_Setting field is the minimum volume output and a value of 255 is the maximum volume output.

### 2.3.1.2 Mute field

The Mute field has two values: Not Muted (or the numeral “0”), and Muted (or the numeral “1”). The Mute field value represents the server’s audio state where a value of Not Muted represents unmuted audio and a value of Muted represents Muted audio.

The Mute field value shall not affect the Volume\_Setting field value; for example, muting a server shall not change the Volume\_Setting field value to 0.

This Mute field provides a single mute point for the entire device. In addition, the AICS Mute field of the Input State characteristic mutes individual inputs.

### 2.3.1.3 Change\_Counter field

The server shall increment the Change\_Counter field value by one upon every change to the Volume\_Setting and Mute field values. The Change\_Counter field should be incremented only once if more than one of these field values are changed at the same time. The Change\_Counter field value is used in all Volume Control Point commands.

The server shall initialize the Change\_Counter field to an arbitrary value. The value shall be in the range of 0 to 255, and an increment past 255 shall roll over to 0.

## 2.3.2 Volume Flags

The Volume Flags characteristic provides additional state information to clients.

### 2.3.2.1 Volume\_Setting\_Persisted field

The Volume\_Setting\_Persisted field informs clients whether the current Volume Setting field value has been set by a user volume change or remains unchanged at the default initialization value as described in Section 3.3.

## 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
Volume State	M	Read, Notify	None	Encryption Required
Volume Control Point	M	Write	None	Encryption Required
Volume Flags	M	Read	Notify	Encryption Required

Table 3.1: VCS characteristics

Properties not listed as Mandatory or Optional in Table 3.1 are Excluded.

### 3.1 Volume State

The Volume State characteristic shall be used to reflect the state of the audio volume to which this service applies. The value of the Volume State characteristic shall follow the format described in Table 3.2.

Field Name	Size (Octets)	Format
Volume_Setting	1	uint8
Mute	1	uint8
Change_Counter	1	uint8

Table 3.2: Volume State characteristic value format

#### 3.1.1 Volume\_Setting field

The Volume\_Setting field shall be used to reflect the volume of the audio to which this service applies.

The Volume\_Setting field is applied as described in Section 2.3.1.1.

#### 3.1.2 Mute field

The Mute field shall be set to a value that reflects the current mute state of the audio to which this service applies.

The Mute field is applied as described in Section 2.3.1.2.

#### 3.1.3 Change\_Counter field

The server shall initialize the Change\_Counter field to an arbitrary value. The Change\_Counter field value shall be incremented by 1 when the Volume\_Setting or Mute field value changes and shall not be changed otherwise. When the Change\_Counter field value reaches 255, its next increment shall be 0.



### 3.1.4 Volume State behavior

The Volume State characteristic value may be read by the client. When the Volume State value changes, the server shall notify clients that have enabled the Client Characteristic Configuration Descriptor for notifications of the new value. The Volume State characteristic value shall be the same for all clients.

## 3.2 Volume Control Point

The Volume Control Point characteristic is used to request a specific procedure to be executed by the server when a value is written to it.

### 3.2.1 Volume Control Point procedure requirements

Table 3.3 lists the requirements for the Volume Control Point procedures for the request opcodes and operands in the context of VCS.

Opcode Value	Opcode	Procedure Section	Opcode Requirement	Operand
0x00	Relative Volume Down	<a href="#">3.2.2.1</a>	M	Change_Counter
0x01	Relative Volume Up	<a href="#">3.2.2.2</a>	M	Change_Counter
0x02	Unmute/Relative Volume Down	<a href="#">3.2.2.3</a>	M	Change_Counter
0x03	Unmute/Relative Volume Up	<a href="#">3.2.2.4</a>	M	Change_Counter
0x04	Set Absolute Volume	<a href="#">3.2.2.5</a>	M	Change_Counter, Volume_Setting
0x05	Unmute	<a href="#">3.2.2.6</a>	M	Change_Counter
0x06	Mute	<a href="#">3.2.2.7</a>	M	Change_Counter

Table 3.3: Volume Control Point procedure requirements

### 3.2.2 Volume Control Point behavior

The Volume Control Point characteristic value may be written by the client.

If a client writes an opcode that is not supported or not defined in [Table 3.3](#), then the server shall return an ATT Error Response with the error code Opcode Not Supported defined in [Table 1.2](#).

If the control point procedure includes the Change\_Counter field, and a client writes a Change\_Counter operand that does not equal the Change\_Counter field of the Volume State characteristic value, then the server shall return an ATT Error Response with the error code Invalid Change Counter defined in [Table 1.2](#).

The Volume Control Point supports two sets of Relative Volume procedures: the first set, Relative Volume Down and Relative Volume Up, does not affect the Mute state of the server, and the second set, Unmute/Relative Volume Down and Unmute/Relative Volume Up, sets the Mute field value to Not Muted.



The server shall choose a positive value called the Step Size for use in the Relative Volume procedures. The Step Size value shall be the same for all Relative Volume procedures.

### 3.2.2.1 Relative Volume Down procedure

If the Relative Volume Down opcode is written to the Volume Control Point and the Change\_Counter operand matches the Change\_Counter field of the Volume State characteristic value, the server shall reduce the value of the Volume\_Setting field through the following equation:

$$\text{Volume\_Setting} = \max(\text{Volume\_Setting} - \text{Step Size}, 0)$$

If the Relative Volume Down procedure causes the Volume\_Setting field value to change, the server shall increment the Change\_Counter field and notify clients of the new Volume State characteristic value, as described in Section 3.1.4.

The Relative Volume Down procedure shall not affect the Mute field value.

The Volume Control Point characteristic value used for the Relative Volume Down procedure shall be formatted as listed in Table 3.4.

Parameter	Size (Octets)	Value
Opcode	1	0x00 = Relative Volume Down Opcode
Change_Counter	1	0x00–0xFF

Table 3.4: Relative Volume Down format

### 3.2.2.2 Relative Volume Up procedure

If the Relative Volume Up opcode is written to the Volume Control Point and the Change\_Counter operand matches the Change\_Counter field of the Volume State characteristic value, the server shall increase the value of the Volume\_Setting field through the following equation:

$$\text{Volume\_Setting} = \min(\text{Volume\_Setting} + \text{Step Size}, 255)$$

If the Relative Volume Up procedure causes the Volume\_Setting field value to change, the server shall increment the Change\_Counter field and notify clients of the new Volume State characteristic value, as described in Section 3.1.4.

The Relative Volume Up procedure shall not affect the Mute field value.

The Volume Control Point characteristic value used for the Relative Volume Up procedure shall be formatted as listed in Table 3.5.

Parameter	Size (Octets)	Value
Opcode	1	0x01 = Relative Volume Up Opcode
Change_Counter	1	0x00–0xFF

Table 3.5: Relative Volume Up format

### 3.2.2.3 Unmute/Relative Volume Down procedure

If the Unmute/Relative Volume Down opcode is written to the Volume Control Point and the Change\_Counter operand matches the Change\_Counter field of the Volume State characteristic value, the server shall reduce the Volume\_Setting field value through the following equation:

$$\text{Volume\_Setting} = \max(\text{Volume\_Setting} - \text{Step Size}, 0)$$

The server shall also set the Mute field value to Not Muted. If the Unmute/Relative Volume Down procedure causes either the Volume\_Setting or Mute field value to change, the server shall increment the value of the Change\_Counter field and notify clients of the new Volume State characteristic value as described in Section 3.1.4.

The Volume Control Point characteristic value used for the Unmute/Relative Volume Down procedure shall be formatted as listed in Table 3.6.

Parameter	Size (Octets)	Value
Opcode	1	0x02 = Unmute/Relative Volume Down Opcode
Change_Counter	1	0x00–0xFF

Table 3.6: Unmute/Relative Volume Down format

### 3.2.2.4 Unmute/Relative Volume Up procedure

If the Unmute/Relative Volume Up opcode is written to the Volume Control Point and the Change\_Counter operand matches the Change\_Counter field of the Volume State characteristic value, the server shall increase the Volume\_Setting field value through the following equation:

$$\text{Volume\_Setting} = \min(\text{Volume\_Setting} + \text{Step Size}, 255)$$

The server shall also set the Mute field value to Not Muted. If the Unmute/Relative Volume Up procedure causes either the Volume\_Setting or Mute field value to change, the server shall increment the value of the Change\_Counter field and notify clients of the new Volume State characteristic value as described in Section 3.1.4.

The Volume Control Point characteristic value used for the Unmute/Relative Volume Up procedure shall be formatted as listed in Table 3.7.

Parameter	Size (Octets)	Value
Opcode	1	0x03 = Unmute/Relative Volume Up Opcode
Change_Counter	1	0x00–0xFF

Table 3.7: Unmute/Relative Volume Up format

### 3.2.2.5 Set Absolute Volume procedure

If the Set Absolute Volume opcode is written to the Volume Control Point and the Change\_Counter operand matches the Change\_Counter field of the Volume State characteristic value, then the server shall set the Volume\_Setting field value to the Volume\_Setting operand value. If the Set Absolute Volume

procedure causes the Volume\_Setting field value to change, the server shall notify clients of the new Volume State value as described in Section 3.1.4.

The Set Absolute Volume procedure shall not affect the Mute field value.

The Volume Control Point characteristic value used for the Set Absolute Volume procedure shall be formatted as listed in Table 3.8.

Parameter	Size (Octets)	Value
Opcode	1	0x04 = Set Absolute Volume Opcode
Change_Counter	1	0x00–0xFF
Volume_Setting	1	0x00–0xFF

Table 3.8: Set Absolute Volume format

### 3.2.2.6 Unmute procedure

If the Unmute opcode is written to the Volume Control Point and the Change\_Counter operand matches the Change\_Counter field of the Volume State characteristic value, the server shall set the Mute field value to Not Muted.

If the Unmute procedure causes the Mute field value to change, the server shall increment the Change\_Counter field and notify clients of the new Volume State characteristic value as described in Section 3.1.4.

The Volume Control Point characteristic value used for the Unmute procedure shall be formatted as listed in Table 3.9.

Parameter	Size (Octets)	Value
Opcode	1	0x05 = Unmute Opcode
Change_Counter	1	0x00–0xFF

Table 3.9: Unmute format

### 3.2.2.7 Mute procedure

If the Mute opcode is written to the Volume Control Point and the Change\_Counter operand matches the Change\_Counter field of the Volume State characteristic value, the server shall set the Mute field value to Muted.

If the Mute procedure causes the Mute field value to change, the server shall increment the Change\_Counter field and notify clients of the new Volume State characteristic value as described in Section 3.1.4.

The Volume Control Point characteristic value used for the Mute procedure shall be formatted as listed in Table 3.10.

Parameter	Size (Octets)	Value
Opcode	1	0x05 = Mute Opcode
Change_Counter	1	0x00–0xFF

Table 3.10: Mute format

### 3.3 Volume Flags

The Volume Flags characteristic shall be set to a value that reflects the properties of VCS.

#### 3.3.1.1 Volume Flags behavior

The Volume Flags characteristic value may be read by the client. If the server supports changing the Volume Flags value, the server shall support notification for the Volume Flags characteristic. When the Volume Flags value changes, the server shall notify clients that have enabled the Client Characteristic Configuration Descriptor for notifications of the new value.

The Volume\_Setting\_Persisted field shall be set to Reset Volume Setting if the Volume\_Setting field value of the Volume State characteristic is an initial value, such as a server reset value, that has not been modified by a user change. A user change includes a client procedure that modifies the Volume\_Setting field value, a server change to the Volume\_Setting field value, or any other modification of the Volume\_Setting field value. The Volume\_Setting\_Persisted field shall be set to User Set Volume Setting if the server does not support changing of the Volume Flags characteristic value.

The Volume\_Setting\_Persisted field shall be set to User Set Volume Setting after a user change occurs to the Volume\_Setting field value of the Volume State characteristic. If a server is reset or disconnected and has persisted the Volume\_Setting field value, then the Volume\_Setting\_Persisted field shall remain set to User Set Volume Setting.

The Volume Flags characteristic value shall have the format described in [Table 3.11](#).

Characteristic Name	Size (Octets)	Format
Volume Flags	1	uint8

Table 3.11: Volume Flags characteristic value format

The fields of the Volume Flags characteristic value shall have the format described in [Table 3.12](#).

Field Name	Bit(s)	Value
Volume_Setting_Persisted	0	0x00 = Reset Volume Setting 0x01 = User Set Volume Setting
RFU	1-7	0x00

Table 3.12: Volume Flags characteristic value fields



## 4 SDP interoperability

If VCS is exposed over Basic Rate/Enhanced Data Rate (BR/EDR), then the service shall have the Service Discovery Protocol (SDP) record defined in [Table 4.1](#).

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	Status
Service Class ID List	–	–	–	M
Service Class #0	–	UUID	«Volume Control Service»	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 if Enhanced Attribute Protocol (EATT), introduced in Volume 3, Part F, Section 3.2.11 in [\[2\]](#), is supported, otherwise Excluded.

## 5 Acronyms and abbreviations

Acronym/Abbreviation	Meaning
AICS	Audio Input Control Service
ATT	Attribute Protocol
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
PDU	Protocol Data Unit
PSM	Protocol/Service Multiplexer
RFU	Reserved for Future Use
SDP	Service Discovery Protocol
UUID	universally unique identifier
VCS	Volume Control Service
VOCS	Volume Offset Control Service

Table 5.1: Acronyms and abbreviations

## 6 References

---

- [1] Bluetooth Core Specification, Version 4.0 or later
- [2] Bluetooth Core Specification, Version 5.2
- [3] Bluetooth Volume Offset Control Service Specification
- [4] Bluetooth Audio Input Control Service Specification
- [5] Bluetooth SIG Assigned Numbers,  
<https://www.bluetooth.com/specifications/assigned-numbers>