

Media Control Service

Bluetooth® Service Specification

- **Version:** v1.0.1
- **Version Date:** 2024-10-01
- **Prepared By:** Generic Audio Working Group

Abstract

This service provides the client with the ability to control and interact with media players.



Version History

Version Number	Date	Comments
v1.0	2021-03-09	Adopted by the Bluetooth SIG Board of Directors.
v1.0.1	2024-10-01	Adopted by the Bluetooth SIG Board of Directors.

Acknowledgments

Name	Company
Robin Heydon	Qualcomm
Tim Reilly	Bose
Michael Rougeux	Bose
Chris Church	Qualcomm
Joel Linsky	Qualcomm
Bjarne Klemmensen	Oticon
Marcel Holtmann	Intel
Nick Hunn	GNResound
Georg Dickmann	Sonova
Asbjørn Sæbø	Nordic Semiconductor
Masahiko Seki	Sony
Daniel Sisolak	Bose
Andrew Estrada	Sony
Scott Walsh	Poly
Rasmus Abildgren	Bose
Frank Yerrace	Microsoft
Nick D'Amato	Sonos
Luiz Von Dentz	Intel
Oren Haggai	Intel

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 © 2018–2024. 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

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 v.1.0.1	7
1.9	Language	8
1.9.1	Language conventions	8
1.9.2	Reserved for Future Use	8
1.9.3	Prohibited	9
2	Service	10
2.1	Description	10
2.1.1	Media player	10
2.1.2	Media player information	11
2.1.3	Current track	11
2.1.4	Source playback speed	11
2.1.5	Seeking	12
2.1.6	Next track	12
2.1.7	Current group	12
2.1.8	Playing order	12
2.1.9	Group and track structure	12
2.1.10	Media state machine	13
2.1.11	Search	13
2.2	Declaration	13
2.3	Included services	13
3	Service characteristics	14
3.1	Media Player Name	15
3.1.1	Characteristic behavior	15
3.2	Media Player Icon Object ID	15
3.2.1	Characteristic behavior	16
3.3	Media Player Icon URL	16
3.3.1	Characteristic behavior	16
3.4	Track Changed	16
3.4.1	Characteristic behavior	16
3.5	Track Title	16
3.5.1	Characteristic behavior	16
3.6	Track Duration	17
3.6.1	Characteristic behavior	17
3.7	Track Position	17
3.7.1	Characteristic behavior	17
3.8	Playback Speed	18
3.8.1	Characteristic behavior	18
3.9	Seeking Speed	18



3.9.1	Characteristic behavior	19
3.10	Current Track Segments Object ID	19
3.10.1	Characteristic behavior	19
3.11	Current Track Object ID	19
3.11.1	Characteristic behavior	19
3.12	Next Track Object ID	20
3.12.1	Characteristic behavior	20
3.13	Parent Group Object ID	20
3.13.1	Characteristic behavior	20
3.14	Current Group Object ID	21
3.14.1	Characteristic behavior	21
3.15	Playing Order	21
3.15.1	Characteristic behavior	22
3.16	Playing Orders Supported	22
3.16.1	Characteristic behavior	23
3.17	Media State	23
3.17.1	Characteristic behavior	24
3.18	Media Control Point	25
3.18.1	Characteristic behavior	26
3.18.2	Media Control Point Notification	30
3.19	Media Control Point Opcodes Supported	31
3.19.1	Characteristic behavior	32
3.20	Search Control Point	32
3.20.1	Characteristic behavior	33
3.20.2	Search Control Point Notification	33
3.21	Search Results Object ID	34
3.21.1	Characteristic behavior	34
3.22	Content Control ID	34
3.22.1	Characteristic behavior	34
4	Object Transfer Service object types	35
4.1	Media Player Icon Object Type	35
4.2	Track Segments Object Type	35
4.2.1	Segment Name Length field	35
4.2.2	Segment Name field	35
4.2.3	Segment Position field	35
4.3	Track Object Type	35
4.4	Group Object Type	36
4.4.1	Object Type field	36
4.4.2	Object ID field	36
5	SDP interoperability	37
6	Acronyms and abbreviations	38
7	References	39

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 optionally depends on the Object Transfer Service (OTS) [3].

1.3 Bluetooth Core Specification release compatibility

This service is compatible with any version of the non-deprecated or non-withdrawn Bluetooth Core Specification.

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 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.

GATT Sub-Procedure	Requirements
Write Characteristic Value	M
Write Without Response	M
Notifications	M
Read Characteristic Descriptors	M
Write Characteristic Descriptors	M

Table 1.1: GATT sub-procedure requirements

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 Core Specification [1].

The server shall return an ATT Error Response with the application error code Value Changed During Read Long (0x80) as defined in Table 1.2 to any ATT_READ_BLOB_REQ with a non-zero Value Offset if the characteristic value has changed since the last zero offset read response of that characteristic value to that client.

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 Attribute Protocol (ATT) bearer. See Volume 3, Part F, Section 3.3.2 in the Bluetooth Core Specification [1].

A higher-layer profile can specify the use of an Enhanced ATT bearer. See Volume 3, Part F, Section 3.3.2 in the Bluetooth Core Specification [1].

1.6 Application error codes

This service defines the ATT Application error codes shown in Table 1.2.

Name	Error Code	Description
Value Changed During Read Long	0x80	A characteristic value has changed while a Read Long Value Characteristic sub-procedure is in progress.

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 Assigned Numbers [2].

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 v.1.0.1

Section	Errata
1.1: Conformance	E23835
1.5: Transport dependencies	E18852
2.1.1: Media Player	E16716
2.3: Included Services	E18999
3.2: Media Player Icon Object ID	E19000
3.18.1: Characteristic behavior	E19003
3.18.1.11: Goto Segment	E19004
3.18.1.16: Goto Track	E19004
3.18.1.21: Goto Group	E19004
3.18.2: Media Control Point Notification	E22507
3.20.1: Characteristic behavior	E19071
3.22.1: Characteristic behavior	E18693

Section	Errata
7: References	E19306

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.

2 Service

2.1 Description

This specification describes two services: Media Control Service (MCS) and Generic Media Control Service (GMCS). Unless otherwise stated in this specification, the behavior of MCS and GMCS is identical, and all the characteristics and the characteristics' behaviors are the same. The term "MCS" is used throughout the document. Unless otherwise specifically stated in this specification, the same meaning applies to GMCS as well.

A device can have multiple logically separate internal media playback entities, referred to in this specification as media players.

GMCS and MCS both expose characteristics that provide status and control of media playback. The GMCS provides status and control of media playback for the device as a single unit. An MCS instance describes and controls the media playback for a specific media player within the device. A device implements MCS instances to allow clients to access the separate internal media playback entities.

For example, a phone or personal computer device can have many end-user applications that play music or movies. The device can:

- Treat these applications as unique media players.
- Implement a GMCS that provides status and control of media playback for the device as a whole.
- Map the status and control to the device's current internal media player or other media playback entity in an implementation-specific manner.
- Implement a separate MCS instance for each media player application, allowing clients to discover, observe, and control each of the device's unique internal media players.

2.1.1 Media player

MCS exposes characteristics that describe a single media player. A media player in this context is a device, or part of a device, that allows another device to control the media that is played. For example, a media player could be a television, a set-top box, a radio, a phone running a radio or music application, a phone running a podcast application, or a similar type of device or application. MCS does not directly manage the audio transport. The description of the media player is comprised of the following:

- Media player information, including media player category, icon, and current rendering targets
- A current track and associated track information
- The next track
- Playback speed
- A current group
- The playing order of tracks within the current group
- The media state of the media player
- The results of a search operation on the media player

2.1.2 Media player information

Each media player should have a name that allows the user to identify the media player. To help identify a media player, an icon may also be exposed. This icon may either be stored locally in an included OTS [3] or be available on the internet at a provided Uniform Resource Locator (URL).

MCS provides characteristics that describe the media player and allows the media player to be controlled.

2.1.3 Current track

The media player has an object called the current track. Some properties of the current track object are exposed through the service as characteristics:

- Track Title
- Current Track Object ID
- Track Position
- Track Duration

The current track object is also exposed through the optionally included OTS and can be accessed by using the Current Track Object ID. The current track always represents the media for which the player is currently playing, paused, or seeking.

If the media player has no current track, then the media player is inactive.

The current track has a track position as well as a track duration. The track position is the current time within the media file from which the current track is being played. The track duration is the length (measured in time) of the media file. The track position can be set to anywhere from the beginning to the end of the media file. It cannot be assumed that the track duration is valid; therefore, the track duration should only be considered a guide to the length of the current track.

When the current track is being played, usually the media is played at the playback speed, which is an adjustable value that may be faster or slower than the nominal, real-time speed of the track.

For streaming media, which does not have a known duration, the duration of the current track has a known reserved value.

Repositioning within the current track can be done relative to the start of the media, relative to the end of the media, or relative to the current track position. Repositioning never moves the track position outside the bounds of the media itself.

To help with repositioning, a track segments object can be exposed that allows different segments within the current track to be discovered. Each segment includes a name and an absolute offset in time from the start of the current track to the start of the segment. Therefore, this offset can be used as an absolute track position to move to the start of a segment.

2.1.4 Source playback speed

The media player can set a playback speed such that media is played at a speed slightly faster or slightly slower than real time. Some media players do not support all the possible playback speeds, and many media players only support playback speeds on a certain subset of tracks. The Playback Speed characteristic exposes the current playback speed.

2.1.5 Seeking

The media player can support fast forward and fast rewind at very high speeds, which is known as the current track seeking speed. During seeking, the current track audio may not be used and only the current track position is updated. The media player can support any seeking speeds. By using the Fast Forward and Fast Rewind opcodes of the Media Control Point, a client can request fast forward or fast rewind multiple times. Each time the server receives a request, the media player may move to a higher fast forward or rewind speed, or the seeking speed may wrap to the lowest speed.

2.1.6 Next track

If a media player is playing content organized by tracks, the server defines which track is the next track in the playing order. By exposing this next track information to the client, the client can pre-load the next track's information so that when the current track changes to the next track, the client can instantly update the information shown to the user.

To enable this, MCS has an object called the next track, which can be exposed by using the Next Track Object ID characteristic. The next track is the track that is played once the current track is finished. If the next track has no value, then the next track is determined by the playing order. If there is no next track, because no additional tracks exist in the playing order after the current track, then the next track is invalid. If the next track is written, then the client is overriding the deterministic next track and selecting a different next track. Once the next track becomes the current track, the normal playing order is used for the next track unless the next track is written again. This allows a client to perform a jukebox style of functionality. Writing the next track may also affect the current group and the playing order when the next track becomes the current track.

2.1.7 Current group

The media player has an object called the current group. Details of the current group are exposed by using the Current Group Object characteristic. The current group is a sequence of object records that exposes the set of tracks, and possibly other groups, within the current group. Information about a group is only available through the included OTS.

The current group is used to determine the next track by using the play order. Only tracks within a single group can be played automatically.

If the current group is changed while the current track is playing, the current track is changed to the first track of the current playing order.

2.1.8 Playing order

The media player has a playing order that determines the next track for a given current track within a current group. The playing order can be any of the following:

- Play a single track or a group of tracks.
- Play tracks in group order, in oldest first order, in newest first order, or shuffle order.
- Play the tracks or groups once or play the tracks or groups repetitively.

2.1.9 Group and track structure

Beyond what is defined in this specification, the organization of group objects and track objects are left up to the implementation. The group and track structure should be set up so that also clients with limited functionality (e.g., clients lacking the functionality to read and write Object IDs) are able to navigate the structure and access all tracks and groups.

In particular, clients that support only the “play” opcode; the “play” and “next track” opcodes; or the “play,” “next track,” and “next group” opcodes should be able to access all groups and tracks available.

2.1.10 Media state machine

The media player has a state machine that represents the states and transitions of the control of the media player. The states include inactive, playing, seeking, and paused.

2.1.11 Search

The media player allows for searches to be performed over the set of tracks and groups. The search results are returned in a search results object that is exposed through the included OTS. The search results object is identical to an ordinary group object and can be selected as the current group.

2.2 Declaration

MCS shall be a «Primary Service» and the service universally unique identifier (UUID) shall be set to «Media Control» as defined in the Bluetooth Assigned Numbers [2].

GMCS shall be a «Primary Service» and the service UUID shall be set to «Generic Media Control» as defined in [2].

2.3 Included services

If any of the Media Player Icon Object ID, Current Track Segments Object ID, Current Track Object ID, Next Track Object ID, Parent Group Object ID, Current Group Object ID, or Search Results Object ID characteristics are exposed by this service, OTS [3] shall be instantiated as a «Secondary Service» and shall be included in MCS or GMCS.

3 Service characteristics

This section defines the characteristic and descriptor requirements for MCS and GMCS. See [Table 3.1](#) for a list of the requirements for MCS and GMCS.

In this specification, UTF-8 strings are not NULL terminated.

Requirements in this section are defined as "Mandatory" (M), "Optional" (O), "Excluded" (E), 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
Media Player Name	M	Read	Notify (C.4)	Encryption required
Media Player Icon Object ID	O	Read	–	Encryption required
Media Player Icon URL	O	Read	–	Encryption required
Track Changed	M	Notify	–	Encryption required
Track Title	M	Read	Notify	Encryption required
Track Duration	M	Read	Notify	Encryption required
Track Position	M	Read, Write, Write Without Response	Notify	Encryption required
Playback Speed	O	Read, Write, Write Without Response	Notify	Encryption required
Seeking Speed	O	Read	Notify	Encryption required
Current Track Segments Object ID	C.1	Read	–	Encryption required
Current Track Object ID	O	Read, Write, Write Without Response	Notify	Encryption required
Next Track Object ID	C.1	Read, Write, Write Without Response	Notify	Encryption required
Parent Group Object ID	C.1	Read	Notify	Encryption required
Current Group Object ID	C.1	Read, Write, Write Without Response	Notify	Encryption required
Playing Order	O	Read, Write, Write Without Response	Notify	Encryption required
Playing Orders Supported	O	Read	–	Encryption required
Media State	M	Read, Notify	–	Encryption required

Characteristic Name	Requirement	Mandatory Properties	Optional Properties	Security Permissions
Media Control Point	O	Write, Write Without Response, Notify	–	Encryption required
Media Control Point Opcodes Supported	C.2	Read	Notify	Encryption required
Search Control Point	C.3	Write, Write Without Response, Notify	–	Encryption required
Search Results Object ID	O	Read, Notify	–	Encryption required
Content Control ID	M	Read	–	Encryption required

Table 3.1: MCS characteristics

- C.1: Mandatory if the Current Track Object ID characteristic is supported, otherwise Excluded
- C.2: Mandatory if the Media Control Point characteristic is supported, otherwise Excluded
- C.3: Mandatory if the Search Results Object ID characteristic is supported, otherwise Excluded
- C.4: Optional for MCS, Mandatory for GMCS

Properties not listed as Mandatory or Optional in the table are Excluded.

3.1 Media Player Name

The Media Player Name characteristic exposes the name of the media player application that is providing media tracks. The characteristic value shall be a UTF-8 string.

3.1.1 Characteristic behavior

The Media Player Name characteristic returns its associated value when it is read by a client.

The Media Player Name characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection. If the characteristic value is greater than (ATT_MTU-3), then the first (ATT_MTU-3) octets shall be included in the notification.

3.2 Media Player Icon Object ID

The Media Player Icon Object ID characteristic is used to access a Media Player Icon Object type as defined in [Section 4.1](#). The Media Player Icon Object ID characteristic identifies the Object ID of an object containing an icon that can identify this media player to the user. If the media player has no icon, the length of the characteristic shall be zero. If the media player has an icon, the length of the characteristic is six octets (the size of uint48) and the value of this characteristic shall be a uint48 value that is in the range of a valid Object ID as defined in OTS. If present, this value can be used in the Object ID characteristic in the included OTS to allow access to the contents of the icon.

3.2.1 Characteristic behavior

The Media Player Icon Object ID characteristic returns its associated value when it is read by a client.

3.3 Media Player Icon URL

The Media Player Icon URL characteristic exposes the URL of an icon that can identify this media player to the user. The value of this characteristic shall be formatted as a UTF-8 string.

3.3.1 Characteristic behavior

The Media Player Icon URL characteristic returns its associated value when read by a client.

3.4 Track Changed

The Track Changed characteristic is used to notify the client when the current track is changed. The Track Changed characteristic is an empty characteristic (size equal to zero).

3.4.1 Characteristic behavior

The Track Changed characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

A server device shall notify this characteristic to the client device depending on the value of the Client Characteristic Configuration descriptor when the track changes.

Any characteristic that will change in the transition between the current track and the next track shall be updated before notifying the Track Changed characteristic.

3.5 Track Title

The Track Title characteristic exposes the title of the current track. The characteristic value shall be a UTF-8 string.

If the media player has no current track or the track title is unavailable, the Track Title characteristic value shall be a zero-length string.

3.5.1 Characteristic behavior

The Track Title characteristic returns its associated value when it is read by a client.

The Track Title characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

A server device shall notify this characteristic to the client device depending on the value of the Client Characteristic Configuration descriptor when the Track Title value is changed. If the characteristic value is longer than (ATT_MTU-3), then the first (ATT_MTU-3) octets shall be included in the notification.

3.6 Track Duration

The Track Duration characteristic exposes the length of the current track in 0.01-second resolution as a 32-bit signed integer.

If the media player has no current track or the duration of the current track is unknown, the Track Duration characteristic value shall be 0xFFFFFFFF. Otherwise, the duration of the track shall be zero or greater.

3.6.1 Characteristic behavior

The Track Duration characteristic returns its associated value when it is read by a client.

If the Track Duration characteristic supports notification, the Track Duration characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

If the Track Duration characteristic supports notification, a server device shall notify this characteristic to the client device when the Track Duration characteristic value changes.

3.7 Track Position

The Track Position characteristic exposes the current track position of the current track in 0.01-second resolution as a 32-bit signed integer. The Track Position characteristic exposes the offset from the start of the track to the current playing position. If the start of the track is not well defined (such as in a live stream), the server sets a starting position (where Track Position equals 0) or sets the value to UNAVAILABLE (0xFFFFFFFF).

The track position of the current track is not used to synchronize audio between multiple devices. It is only used to display the current track position to the user and to allow the track position to be quickly repositioned (for example, to support client bookmarks). If the media player has no current track, the Track Position characteristic value shall be 0xFFFFFFFF.

3.7.1 Characteristic behavior

The Track Position characteristic returns its associated value when it is read by a client.

When the Track Position characteristic is written, the server may adjust the current playing position of the current track. If the value is zero or greater, then the current playing position shall be set to the offset from the start of the track. If the value is less than zero, then the current playing position shall be set to the offset from the end of the track and the value of the Track Position characteristic shall be set to the offset from the start of the track to the new playing position. If the value written does not correspond to a valid track position, the server shall set the Track Position characteristic to a valid value.

If the Track Position characteristic supports notification, the Track Position characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

If the Track Position characteristic supports notification, a server device shall notify this characteristic to the client device based on the value of the Client Characteristic Configuration descriptor when the Track Position characteristic value is changed, when the track is paused, or when the playback speed is changed. As an exception to this, to avoid an excessive number of notifications, the Track Position should not be notified when the Media State is set to "Playing" and playback happens at a constant speed. The

Track Position should be notified when the Media State characteristic is set to “Seeking”. The notification frequency while seeking is determined by the server implementation.

3.8 Playback Speed

The Playback Speed characteristic exposes the current track playback speed. The value of this characteristic shall be a signed 8-bit integer (p) that is used to compute the multiple of the playback speed (s) by calculating the value of 2 to the power of p divided by 64:

$$s = 2^{\frac{p}{64}}$$

The equation results in playback speeds from .25x normal speed up to nearly 4x normal speed, as shown in [Table 3.2](#).

p – Characteristic Value	s – Resultant Multiple of Playback Speed
-128	.25
-64	.50
0	1
64	2
127	3.957 (nearly 4)

Table 3.2: A few examples of playback speed derived values

3.8.1 Characteristic behavior

The Playback Speed characteristic returns its associated value (p) when read by a client.

When the Playback Speed characteristic is written, the server shall adjust the playback speed of the current track. If the server supports the value written, the server shall set the Playback Speed characteristic to the value written. If the server does not support the value written, the server shall set the Playback Speed characteristic to a supported value. If the value written is not supported and is greater than the existing Playback Speed characteristic, then the server should set the Playback Speed characteristic to the next higher supported playback speed. If the value written is not supported and is less than the existing Playback Speed characteristic, then the server should set the Playback Speed characteristic to the next lower supported playback speed.

For streaming media, the playback speed may be fixed to 1x normal speed.

If the Playback Speed characteristic supports notification, the Playback Speed characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

If the Playback Speed characteristic supports notification, a server device shall notify this characteristic to the client device depending on the value of the Client Characteristic Configuration descriptor when the Playback Speed characteristic value is changed or written.

3.9 Seeking Speed

The Seeking Speed characteristic exposes the seeking speed of the current track. The value of this characteristic shall be a signed 8-bit integer, which is the current multiple of the real-time playback speed

at which the track is being fast forwarded or fast rewind. (“Real-time” is the point at which $p=0$ in the Playback Speed characteristic calculation.) Fast forward is a positive current track-seeking speed. Fast rewind is a negative current track-seeking speed. If the media player is not seeking, the value of this characteristic shall be 0. During seeking, the current track audio may not be used, and the track position is handled as described in [Section 3.7.1](#).

3.9.1 Characteristic behavior

The Seeking Speed characteristic returns its associated value when it is read by a client.

If the Seeking Speed characteristic supports notification, the Seeking Speed characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

If the Seeking Speed characteristic supports notification, a server device shall notify this characteristic to the client device depending on the value of the Client Characteristic Configuration descriptor when the Seeking Speed characteristic value is changed.

3.10 Current Track Segments Object ID

The Current Track Segments Object ID characteristic is used to access a Track Segment Object type as defined in [Section 4.2](#). The Current Track Segments Object ID characteristic identifies the segments of the current track as an object in the included OTS. If the media player has no current track, the length of the characteristic shall be zero. If the media player has a current track, the length of the characteristic is six octets (the size of uint48) and the value of this characteristic shall be a uint48 value that is in the range of a valid Object ID as defined in OTS. If present, the uint48 value can be used in the Object ID characteristic in the included OTS to allow access to the current track segments information of the current track.

The object in the OTS that exposes the track segments information can be reused for each track.

3.10.1 Characteristic behavior

The Current Track Segments Object ID characteristic returns its associated value when it is read by a client.

3.11 Current Track Object ID

The Current Track Object ID characteristic is used to access a Track Object type as defined in [Section 4.3](#). This characteristic identifies the current track as an object in the included OTS. If the media player has no current track, the length of the characteristic shall be zero. If the media player has a current track, the length of the characteristic is six octets (the size of uint48) and the value of this characteristic shall be a uint48 value that is in the range of a valid Object ID as defined in OTS. If present, this value can be used in the Object ID characteristic in the included OTS to allow access to the contents of the track, such as the embedded artwork for the track.

3.11.1 Characteristic behavior

The Current Track Object ID characteristic returns its associated value when it is read by a client.

When the Current Track Object ID characteristic is written, the server shall change the current track to that identified by the value and act as described in the Goto Track command in [Section 3.18.1.16](#).

If the Current Track Object ID characteristic supports notification, the Current Track Object ID characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

If the Current Track Object ID characteristic supports notification, a server device shall notify this characteristic to the client device when the Current Track Object ID characteristic value changes, except when the client changes the value.

3.12 Next Track Object ID

The Next Track Object ID characteristic is used to access a Track Object type as defined in [Section 4.3](#). This characteristic identifies the next track, as defined by the playing order and current group object, as an object in the included OTS. If the media player has no next track, the length of the characteristic shall be zero. If the media player has a next track, the length of the characteristic is six octets (the size of uint48) and the value of this characteristic shall be a uint48 value that is in the range of a valid Object ID as defined in OTS. If present, this value can be used in the Object ID characteristic in the included OTS to allow access to the contents of the track, such as the embedded artwork for the track.

3.12.1 Characteristic behavior

The Next Track Object ID characteristic returns its associated value when it is read by a client.

When the Next Track Object ID characteristic is written, the server shall change the next track to that identified by the value written to the characteristic by the client.

If the Next Track Object ID characteristic supports notification, the Next Track Object ID characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

If the Next Track Object ID characteristic supports notification, a server device shall notify this characteristic to the client device when the Next Track Object ID characteristic value changes, except when the change is because of the client writing the value to the characteristic.

3.13 Parent Group Object ID

The Parent Group Object ID characteristic is used to access a Group Object type as defined in [Section 4.4](#). This characteristic identifies the parent group of the current group object as a group object in the included OTS. If the media player has not created a parent group or a current group, the length of the characteristic shall be zero. If the media player has a parent group, the length of the characteristic is six octets (the size of uint48) and the value of this characteristic shall be a uint48 value that is in the range of a valid Object ID as defined in OTS. If present, the value can be used in the Object ID characteristic in the included OTS to allow access to the contents of the current parent group.

For a group that has no parent, the value of the Parent Group Object ID shall be the value of the Current Group Object ID.

The value of the Parent Group Object ID can change when the Current Group Object ID changes.

3.13.1 Characteristic behavior

The Parent Group Object ID characteristic returns its associated value when it is read by a client.



The Parent Group Object ID characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

A server device shall notify this characteristic to the client device when the Parent Group Object ID characteristic value changes.

3.14 Current Group Object ID

The Current Group Object ID characteristic is used to access a Group Object type as defined in [Section 4.4](#). This characteristic identifies the current group as a group object in the included OTS. If the media player has not created a current group, the length of the characteristic shall be zero. If the media player has a current group, the length of the characteristic is six octets (the size of uint48) and the value of this characteristic shall be a uint48 value that is in the range of a valid Object ID as defined in OTS. If present, the value can be used in the Object ID characteristic in the included OTS to allow access to the tracks and/or other groups of a group.

3.14.1 Characteristic behavior

The Current Group Object ID characteristic returns its associated value when it is read by a client.

When the Current Group Object ID characteristic is written, the server shall change the current group and shall set the Current Track to the first track of the group according to the current playing order and the Track Position shall be set to 0.

If the Current Group Object ID characteristic supports notification, the Current Group Object ID characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

If the Current Group Object ID characteristic supports notification, a server device shall notify this characteristic to the client device when the Current Group Object ID characteristic value changes, except when the change is because of the client writing the value.

See [Section 2.1.7](#) for a further description of current group.

3.15 Playing Order

The Playing Order characteristic exposes the playing order of the media player. The value of this characteristic is an enumerated value as defined in [Table 3.3](#). The resultant playing order when referencing “Oldest” and “Newest” in the “Name” column in [Table 3.3](#) is implementation-specific.

Value	Name	Description
0x01	Single once	A single track is played once; there is no next track.
0x02	Single repeat	A single track is played repeatedly; the next track is the current track.
0x03	In order once	The tracks within a group are played once in track order.
0x04	In order repeat	The tracks within a group are played in track order repeatedly.

Value	Name	Description
0x05	Oldest once	The tracks within a group are played once only from the oldest first.
0x06	Oldest repeat	The tracks within a group are played from the oldest first repeatedly.
0x07	Newest once	The tracks within a group are played once only from the newest first.
0x08	Newest repeat	The tracks within a group are played from the newest first repeatedly.
0x09	Shuffle once	The tracks within a group are played in random order once.
0x0A	Shuffle repeat	The tracks within a group are played in random order repeatedly.
All other values	RFU	Reserved for Future Use

Table 3.3: Playing Order characteristic values

3.15.1 Characteristic behavior

The Playing Order characteristic returns its associated value when it is read by a client.

When the Playing Order characteristic is written, the playing order should be changed to the value written.

If the value written is not supported, the value written is ignored and the current value does not change.

If the Playing Order characteristic supports notification, the Playing Order characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

If the Playing Order characteristic supports notification, a server device shall notify this characteristic to the client device when the Playing Order characteristic value changes.

The overall playing order is left up to the media player implementation. For example, if the end of a group is reached, the media player may continue to play from another group.

3.16 Playing Orders Supported

The Playing Orders Supported characteristic is a 16-bit bit field representing the supported playing orders described in Table 3.4. The bit field contains all the playing orders that the player supports.

Value	Name	Description
0x0001	Single once	A single track is played once; there is no next track.
0x0002	Single repeat	A single track is played repeatedly; the next track is the current track.
0x0004	In order once	The tracks within a group are played once in track order.



Value	Name	Description
0x0008	In order repeat	The tracks within a group are played in track order repeatedly.
0x0010	Oldest once	The tracks within a group are played once only from the oldest first.
0x0020	Oldest repeat	The tracks within a group are played from the oldest first repeatedly.
0x0040	Newest once	The tracks within a group are played once only from the newest first.
0x0080	Newest repeat	The tracks within a group are played from the newest first repeatedly.
0x0100	Shuffle once	The tracks within a group are played in random order once.
0x0200	Shuffle repeat	The tracks within a group are played in random order repeatedly.
0x0400 – 0x8000	RFU	Reserved for Future Use

Table 3.4: Playing Orders Supported characteristic bit values

3.16.1 Characteristic behavior

The Playing Orders Supported characteristic returns its associated value when it is read by a client.

3.17 Media State

The Media State characteristic is defined as an 8-bit enumerated value. The following states are defined in Table 3.5.

Opcode Value	Name	Description
0x00	Inactive	The current track is invalid, and no track has been selected.
0x01	Playing	The media player is playing the current track.
0x02	Paused	The current track is paused. The media player has a current track, but it is not being played.
0x03	Seeking	The current track is fast forwarding or fast rewinding.
0x04-0xFF	RFU	Reserved for Future Use

Table 3.5: Media State characteristic values

The media player has a state machine that represents the states and transitions of the control of the media player. The states include Inactive, Playing, Paused, and Seeking, as shown in Figure 3.1.

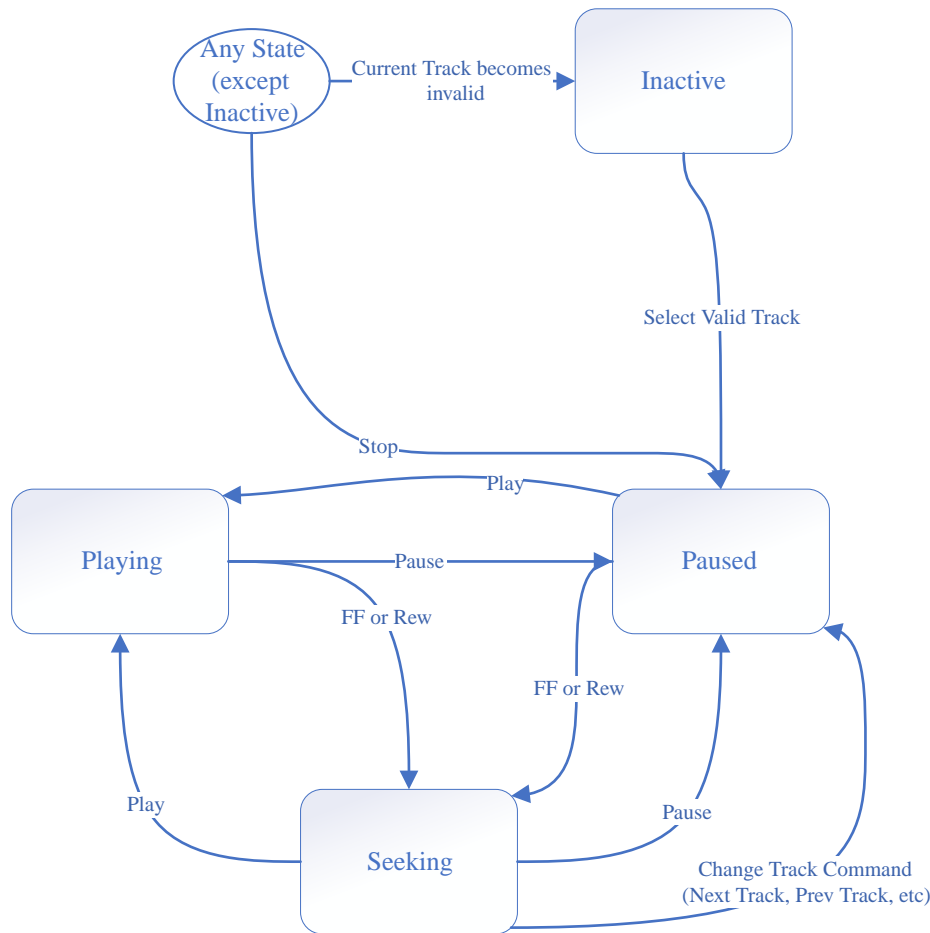


Figure 3.1: Example of a media state machine transition diagram

These states do not relate to any audio streams that are used to play the audio. It is possible for the media player to be in the Playing state without a related audio stream, or in the Inactive and Paused state with a related audio stream. A media player should manage the state of the related audio stream to match the state of the media player, but this relationship is not defined within this service and is left to the implementation. **E16716**

3.17.1 Characteristic behavior

The Media State characteristic returns its associated value when it is read by a client by using the GATT Read Characteristic Value sub-procedure.

The Media State characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

A server device shall notify this characteristic to the client device when the Media State characteristic value changes.

3.18 Media Control Point

The Media Control Point characteristic is defined as an 8-bit enumerated value, known as the opcode, followed by zero or more parameter octets. The opcodes are defined in [Table 3.6](#).

Opcode	Name	Requirement	Description
0x01	Play	C.1	Start playing the current track.
0x02	Pause	C.1	Pause playing the current track.
0x03	Fast Rewind	C.1	Fast rewind the current track.
0x04	Fast Forward	C.1	Fast forward the current track.
0x05	Stop	C.1	Stop current activity and return to the paused state and set the current track position to the start of the current track.
0x10	Move Relative	C.1	Set a new current track position relative to the current track position.
0x20	Previous Segment	C.1	Set the current track position to the starting position of the previous segment of the current track.
0x21	Next Segment	C.1	Set the current track position to the starting position of the next segment of the current track.
0x22	First Segment	C.1	Set the current track position to the starting position of the first segment of the current track.
0x23	Last Segment	C.1	Set the current track position to the starting position of the last segment of the current track.
0x24	Goto Segment	C.1	Set the current track position to the starting position of the n^{th} segment of the current track.
0x30	Previous Track	C.1	Set the current track to the previous track based on the playing order.
0x31	Next Track	C.1	Set the current track to the next track based on the playing order.
0x32	First Track	C.1	Set the current track to the first track based on the playing order.
0x33	Last Track	C.1	Set the current track to the last track based on the playing order.
0x34	Goto Track	C.1	Set the current track to the n^{th} track based on the playing order.

Opcode	Name	Requirement	Description
0x40	Previous Group	C.1	Set the current group to the previous group in the sequence of groups.
0x41	Next Group	C.1	Set the current group to the next group in the sequence of groups.
0x42	First Group	C.1	Set the current group to the first group in the sequence of groups.
0x43	Last Group	C.1	Set the current group to the last group in the sequence of groups.
0x44	Goto Group	C.1	Set the current group to the n th group in the sequence of groups.
All other values	RFU	–	Reserved for Future Use

Table 3.6: Media Control Point characteristic values

C.1: Optional, but Mandatory to support at least one of the opcodes

3.18.1 Characteristic behavior

When the Media Control Point characteristic is written, the server shall perform the behavior defined by the first octet of that value.

If a control point is written with an opcode that is RFU or is not supported by the server, the server shall send the Media Control Point Notification with the OP CODE NOT SUPPORTED Result Code.

If the requested action of any control point write cannot be completed successfully because of an error condition within the player (a condition that is not because the player is in the Inactive state), the server shall send a Media Control Point notification with the COMMAND CANNOT BE COMPLETED Result Code.

If the Media State is Inactive when a control point write occurs, the server shall do one of the following:

- Send a Media Control Point Notification with the MEDIA PLAYER INACTIVE Result Code and not perform the requested action of the control point write.
- Perform, or attempt to perform, the requested action of the control point write and set the Media State to the appropriate state and respond with the appropriate Media Control Point Notification.

3.18.1.1 Play

The Play opcode has no parameters. If the Media State characteristic value is Paused or Seeking, the media player shall start playing the current track and the Media State characteristic value shall be set to Playing.

If the Media State characteristic value is Playing, the Media State characteristic value remains set to Playing.

3.18.1.2 Pause

The Pause opcode has no parameters. If the Media State characteristic value is Playing, the media player shall pause playing the current track and the Media State characteristic value shall be set to Paused.

If the Media State characteristic value is Seeking, the media player shall stop seeking, set the current track and track position as a result of seeking, and the Media State characteristic value shall be set to Paused.

If the Media State characteristic value is Paused, the Media State characteristic value remains set to Paused.

3.18.1.3 Fast Rewind

The Fast Rewind opcode has no parameters. If the Media State characteristic is Paused or Playing, the media player shall move the current track position backwards. The media player should stop playing the current track, set the Media State characteristic to Seeking, and move the current track position backwards at a speed determined by the implementation.

If the Media State characteristic value is Seeking, the seeking speed should be adjusted in the negative direction. The method by which the media player handles multiple Fast Rewind opcode writes is left up to the implementation.

If supported, the Seeking Speed characteristic shall be updated as described in [Section 3.9](#).

3.18.1.4 Fast Forward

The Fast Forward opcode has no parameters. If the Media State characteristic value is Paused or Playing, the media player shall move the current track position forward. The media player should stop playing the current track, set the Media State characteristic to Seeking, and move the current track position forward at a speed determined by the implementation.

If the Media State characteristic is Seeking, the seeking speed should be adjusted in the positive direction. The method by which the media player handles multiple Fast Forward opcode writes is left up to the implementation.

If supported, the Seeking Speed characteristic shall be updated as described in [Section 3.9](#).

3.18.1.5 Stop

The Stop opcode has no parameters. If the Media State characteristic value is Playing, Paused, or Seeking, the media player shall stop any activity and the Media State characteristic shall be set to Paused. The track position shall be set to the beginning of the current track.

3.18.1.6 Move Relative

The Move Relative opcode has a 32-bit signed integer parameter called the “offset”. If the Media State characteristic value is Playing, Paused, or Seeking, the track position of the current track shall be set to the current track position in addition to the offset. The current track position shall not move before the start of the track and shall not move past the end of the track. The current track position shall not wrap. If the addition of the offset to the current track position results in a negative track position, the Track Position characteristic shall be set to 0. If the addition of the offset to the current track position results in a track position past the end of the current track, the Track Position characteristic shall be set to the end of the track.

3.18.1.7 Previous Segment

The Previous Segment opcode has no parameters. If the Media State characteristic value is Playing, Paused, or Seeking, the current track position shall be set to either the starting position of the previous segment of this track or the starting position of the current segment, as determined by the implementation. The previous segment is defined as the segment with a segment position less than the current track position. The comparison should be to a reasonable user margin of error such that if the Previous Segment opcode is used within a few seconds of the start of a segment, then the previous segment is the segment before the current segment, but after a few seconds, the previous segment is the start of the current segment. The exact time used in the margin of error is an implementation detail.

If the opcode is received while in the first segment, the current track position shall be set to the starting position of the first segment.

The resulting Media State characteristic value is left up to the implementation.

3.18.1.8 Next Segment

The Next Segment opcode has no parameters. If the Media State characteristic value is Playing, Paused, or Seeking, the current track position shall be set to the starting position of the next segment of this track. The next segment is defined as the segment with a segment position greater than the current position within the current track. If the opcode is received while in the last segment, the current track position shall be set to the position of the end of the segment.

The resulting Media State characteristic value is left up to the implementation.

3.18.1.9 First Segment

The First Segment opcode has no parameters. If the Media State characteristic value is Playing, Paused, or Seeking, the current track position shall be set to the starting position of the first segment of this track.

The resulting Media State characteristic value is left up to the implementation.

3.18.1.10 Last Segment

The Last Segment opcode has no parameters. If the Media State characteristic value is Playing, Paused, or Seeking, the current track position shall be set to the starting position of the last segment of this track.

The resulting Media State characteristic value is left up to the implementation.

3.18.1.11 Goto Segment

The Goto Segment opcode has a single 32-bit signed integer parameter called “ n ”. If the Media State characteristic value is Playing, Paused, or Seeking, the current track position shall be set to the starting position of the n^{th} segment within the current track. If the value n is a positive number, setting the current track position is equivalent to sending the First Segment opcode once, followed by the Next Segment opcode $|n| - 1$ times. If the value n is a negative number setting, the current track position is equivalent to sending the Last Segment opcode once, followed by the Previous Segment opcode $|n| - 1$ times. If the value of n is zero, the current track position shall not change.

The resulting Media State characteristic value is left up to the implementation.

3.18.1.12 Previous Track

The Previous Track opcode has no parameters. If the Media State characteristic value is Playing, Paused, or Seeking, the current track shall be set to either the previous track based on the playing order

or remain on the current track, as determined by the implementation. The Track Position characteristic shall be set to 0. The resulting Media State characteristic value is left up to the implementation.

3.18.1.13 Next Track

The Next Track opcode has no parameters. If the Media State characteristic value is Playing, Paused, or Seeking, the current track shall be set to the next track based on the playing order and the Track Position shall be set to 0. The resulting Media State characteristic value is left up to the implementation.

3.18.1.14 First Track

The First Track opcode has no parameters. If the Media State characteristic value is Playing, Paused, or Seeking, the current track shall be set to the first track based on the playing order and the Track Position shall be set to 0. The resulting Media State characteristic value is left up to the implementation.

3.18.1.15 Last Track

The Last Track opcode has no parameters. If the Media State characteristic value is Playing, Paused, or Seeking, the current track shall be set to the last track based on the playing order and the Track Position shall be set to 0. The resulting Media State characteristic value is left up to the implementation.

3.18.1.16 Goto Track

The Goto Track opcode has a single 32-bit signed integer parameter called " n ". If the Media State characteristic value is Playing, Paused, or Seeking, the current track shall be set to the n^{th} track based on the playing order. If the value n is a positive number, setting the current track is equivalent to sending the First Track opcode once, followed by the Next Track opcode $|n| - 1$ times. If the value n is a negative number, then setting the current track is equivalent to sending the Last Track opcode once, followed by the Previous Track opcode $|n| - 1$ times. If the value n is zero, the current track shall not be changed. In all cases, the Track Position shall be set to 0. The resulting Media State characteristic value is left up to the implementation.

3.18.1.17 Previous Group

The Previous Group opcode has no parameters. If the Media State characteristic value is Playing, Paused, or Seeking, the current group shall be set to the previous group in the current parent group. The current track shall be set to the first track of the current group resulting from the Previous Group command and the Track Position shall be set to 0. The resulting Media State characteristic value is left up to the implementation.

3.18.1.18 Next Group

The Next Group opcode has no parameters. If the Media State characteristic value is Playing, Paused, or Seeking, the current group shall be set to the next group in the current parent group. The current track shall be set to the first track of the current group resulting from the Next Group command and the Track Position shall be set to 0. The resulting Media State characteristic value is left up to the implementation.

3.18.1.19 First Group

The First Group opcode has no parameters. If the Media State characteristic value is Playing, Paused, or Seeking, the current group shall be set to the first group in the current parent group. If the current group is already the first group within the current parent group, the current group shall not be changed. The current track shall be set to the first track of the current group resulting from the First Group command

and the Track Position shall be set to 0. The resulting Media State characteristic value is left up to the implementation.

3.18.1.20 Last Group

The Last Group opcode has no parameters. If the Media State characteristic value is Playing, Paused, or Seeking, the current group shall be set to the last group in the current parent group. If the current group is already the last group within the current parent group, the current group shall not be changed. The current track shall be set to the first track of the current group resulting from the Last Group command and the Track Position shall be set to 0. The resulting Media State characteristic value is left up to the implementation.

3.18.1.21 Goto Group

The Goto Group opcode has a single 32-bit signed integer parameter called “ n ”. If the Media State characteristic value is Playing, Paused, or Seeking, the current group shall be set to the n^{th} group of the current parent group. If the value n is a positive number, setting the current group is equivalent to sending the First Group opcode once, followed by the Next Group opcode $|n| - 1$ times. If the value n is a negative number, setting the current group is equivalent to sending the Last Group opcode once, followed by the Previous Group opcode $|n| - 1$ times. The current track shall be set to the first track of the current group resulting from the Goto Group command and the Track Position shall be set to 0. If the value n is zero, the current group shall not be changed. The resulting Media State characteristic value is left up to the implementation.

3.18.2 Media Control Point Notification

The Media Control Point characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

Table 3.7 shows the required structure of the Media Control Point Notification. The Media Control Point Notification is sent from the server to the client after each Media Control Point opcode write. If the requested opcode procedure was successfully completed, then the Result Code of the Media Control Point Notification shall be set to SUCCESS. If the requested opcode procedure was not successfully completed, the Result Code shall be set to a value from Table 3.8.

Field	Size	Description
Requested Opcode	1 octet	Opcode written (defined in Table 3.6)
Result Code	1 octet	Specific code as a result of the opcode write

Table 3.7: Media Control Point Notification format

Result Code	Definition	Description
0x01	SUCCESS	Action requested by the opcode write was completed successfully.
0x02	OPCODE NOT SUPPORTED	An invalid or unsupported opcode was used for the Media Control Point write.

Result Code	Definition	Description
0x03	MEDIA PLAYER INACTIVE	The Media State characteristic value is Inactive when the opcode is received or the result of the requested action of the opcode results in the Media State characteristic being set to Inactive.
0x04	COMMAND CANNOT BE COMPLETED	The requested action of any Media Control Point write cannot be completed successfully because of a condition within the player.
All other values	RFU	Reserved for Future Use

Table 3.8: Media Control Point Notification Result Codes

3.19 Media Control Point Opcodes Supported

The Media Control Point Opcodes Supported characteristic exposes the currently supported media control opcodes on the server. The value of this characteristic is a bit field of 32 bits, where bit n is set to 1 when opcode n is supported; otherwise, the bit is set to 0. The opcodes and bit fields are shown in Table 3.9.

Supported Value	Name
0x00000001	Play
0x00000002	Pause
0x00000004	Fast Rewind
0x00000008	Fast Forward
0x00000010	Stop
0x00000020	Move Relative
0x00000040	Previous Segment
0x00000080	Next Segment
0x00000100	First Segment
0x00000200	Last Segment
0x00000400	Goto Segment
0x00000800	Previous Track
0x00001000	Next Track
0x00002000	First Track
0x00004000	Last Track
0x00008000	Goto Track
0x00010000	Previous Group

Supported Value	Name
0x00020000	Next Group
0x00040000	First Group
0x00080000	Last Group
0x00100000	Goto Group
0x00200000 – 0x80000000	RFU

Table 3.9: Media Control Point Opcodes Supported characteristic bit values

3.19.1 Characteristic behavior

The Media Control Point Opcodes Supported characteristic returns its associated value when it is read by a client.

The Media Control Point Opcodes Supported characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

A server device shall notify this characteristic to the client device when the set of supported opcodes changes.

3.20 Search Control Point

The structure of the Search Control Point characteristic is defined as a sequence of search control items. The maximum size of this characteristic value is 64 octets.

Each search control item is composed of a Length field, a Type field, and a Parameter field, as shown in Table 3.10.

Field	Size	Description
Length	1 octet	Total length of the Type and Parameter fields
Type	1 octet	Determines the meaning of the Parameter field
Parameter	Length – 1 octet	Contains the parameter or parameters as defined by the Type field

Table 3.10: Search Control Point search control item fields

The search control items are concatenated into a single sequence of octets. This sequence of octets can be written into the Search Control Point characteristic to initiate a search operation.

The Type field values are defined in Table 3.11.

Type Value	Name	Parameter Description
0x01	Track Name	UTF-8 string
0x02	Artist Name	UTF-8 string

Type Value	Name	Parameter Description
0x03	Album Name	UTF-8 string
0x04	Group Name	UTF-8 string
0x05	Earliest Year	UTF-8 string
0x06	Latest Year	UTF-8 string
0x07	Genre	UTF-8 string
0x08	Only Tracks	None
0x09	Only Groups	None
0x0A-0xFF	RFU	Reserved for Future Use

Table 3.11: Search Control Point item type value parameters

3.20.1 Characteristic behavior

When the Search Control Point characteristic is written, the server shall start a search operation and populate the current or new search results object. When the search results object is fully populated, the server shall notify the Search Results Object ID characteristic to the client device if notifications are enabled.

The value written to the Search Control Point characteristic is a sequence of zero or more search control items. Each search control item logically generates a set of tracks and groups. The set of tracks and groups included in the search results object is a set of tracks and groups that satisfy all of the provided search control items. If zero search control point items are written to the Search Control Point, the result of the search is left up to the implementation.

For example, to discover all the tracks from the year 1999, a search would be performed for "only tracks," with the "earliest year" of "1999," and the "latest year" of "1999."

If no set of tracks or groups are generated with the provided search criteria, there are no search results and the Search Results Object ID characteristic shall be set to a length of 0.

The order of search results is not defined by this specification.

3.20.2 Search Control Point Notification

The Search Control Point characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

Table 3.12 shows the required structure of the Search Control Point Notification. The Search Control Point Notification is sent from the server to the client after each Search Control Point opcode write. If the requested opcode procedure was successfully completed, then the Result Code of the Search Control Point Notification shall be set to SUCCESS. If the requested opcode procedure was not successfully completed, the Result Code should be set to a value from Table 3.13.

Field	Size	Description
Result Code	1 octet	Specific code as a result of the search control point write

Table 3.12: Search Control Point Notification format

Result Code	Definition	Description
0x01	SUCCESS	Search request was accepted; search has started.
0x02	FAILURE	Search request was invalid; no search started.

Table 3.13: Search Control Point Notification Result Codes

3.21 Search Results Object ID

The Search Results Object ID characteristic is used to access a Group Object type as defined in [Section 4.4](#). This characteristic identifies the search results object as a group object in the included OTS. If there are no search results, the length of the characteristic shall be zero. If there are search results, the length of the characteristic is six octets (the size of uint48) and the value of this characteristic shall be a uint48 value that is in the range of a valid Object ID as defined in OTS. If present, the uint48 value can be used in the Object ID characteristic in the included OTS to allow access to the objects in the search results.

3.21.1 Characteristic behavior

The Search Results Object ID characteristic returns its associated value when it is read by a client.

The Search Results Object ID characteristic can be configured for notification by using the GATT Write Characteristic Descriptors sub-procedure on the Client Characteristic Configuration descriptor. When configured for notification, this characteristic can be notified while in a connection.

When the search results object is fully populated, the server shall notify the Search Results Object ID characteristic to the client device if notifications are enabled.

3.22 Content Control ID

The server shall expose the Content Control ID (CCID) characteristic as defined in [\[5\]](#).

3.22.1 Characteristic behavior

The Content Control ID characteristic returns its associated value when read by a client.

The CCID characteristic value in an instance of the service may change only if the handle range for that service is affected by a service change. Otherwise, the CCID characteristic value shall not change and shall persist across connections.

There shall only be one instance of the CCID characteristic in an instance of MCS.

4 Object Transfer Service object types

4.1 Media Player Icon Object Type

The Media Player Icon Object Type shall be identified with the <<Media Player Icon Object Type>> UUID. The format should be in .png format [6].

4.2 Track Segments Object Type

The Track Segments Object Type shall be identified with the «Track Segments Object Type» UUID. The format of the track segments object is a sequence of segment labels, ordered by time. Each segment label has the following format defined in Table 4.1.

Field	Data Type	Size (octets)	Description
Segment Name Length	uint8	1	See Section 4.2.1.
Segment Name	utf8s	Variable	See Section 4.2.2.
Segment Position	sint32	4	See Section 4.2.3.

Table 4.1: Track Segments Object Type format

4.2.1 Segment Name Length field

The Segment Name Length field is an unsigned 8-bit integer that determines the length of the Segment Name in octets.

4.2.2 Segment Name field

The Segment Name field is a sequence of UTF-8 octets that is n octets long, where n is the value of Segment Name Length.

4.2.3 Segment Position field

The Segment Position field is a time offset from the start of the current track to the start of the segment in 0.01-second intervals represented as a 32-bit signed integer. The value of the Segment Position field shall be 0 or greater.

4.3 Track Object Type

The Track Object Type shall be identified with the «Track Object Type» UUID. The format of the Track Object is a uint48 ID from OTS that identifies a single track. A single track should be in the ID3v2 format [4].

4.4 Group Object Type

The Group Object Type shall be identified with the «Group Object Type» UUID. The format of the Group Object is a sequence of records. Each record has the format defined in [Table 4.2](#).

Field	Data Type	Size (octets)	Description
Object Type	uint8	1	See Section 4.4.1 .
Object ID	uint48	6	See Section 4.4.2 .

Table 4.2: Group Object Type format

4.4.1 Object Type field

The enumeration of the Object Type field is defined in [Table 4.3](#).

Description	Value
Track	0
Group	1
RFU	2-255

Table 4.3: Object Type field

4.4.2 Object ID field

The Object ID field is a uint48 ID from OTS which ID identifies a Track Object (described in [Section 4.3](#)) or another Group Object.

5 SDP interoperability

For services that are exposed over Base Rate/Enhanced Data Rate (BR/EDR), include a Service Discovery Protocol (SDP) Record table, as shown in [Table 5.1](#).

Requirements in this section are defined as “Mandatory” (M), “Optional” (O), “Excluded” (E), 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	« Generic Media Control »	M
Service Class #1	–	UUID	« Media Control »	C.1
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.2
Protocol Descriptor List	–	Data Element Sequence	–	C.2
Protocol #0	–	UUID	«L2CAP»	C.2
Parameter #0 for Protocol #0	PSM	uint16	PSM = EATT	C.2
Protocol #1	–	UUID	«ATT»	C.2
BrowseGroupList	–	–	PublicBrowseRoot*	M

Table 5.1: SDP Record

* PublicBrowseRoot shall be present; however, other browse UUIDs may also be included in the list.

C.1: Mandatory if MCS is supported, otherwise Excluded

C.2: Mandatory if Enhanced Attribute Protocol (EATT) is supported, otherwise Excluded

6 Acronyms and abbreviations

Acronym/Abbreviation	Meaning
ATT	Attribute Protocol
BR/EDR	Basic Rate/Enhanced Data Rate
CCID	Content Control ID
EATT	Enhanced Attribute Protocol
GATT	Generic Attribute Profile
GMCS	Generic Media Control Service
L2CAP	Logical Link Control and Adaptation Protocol
LSO	least significant octet
MCS	Media Control Service
OTS	Object Transfer Service
PDU	Protocol Data Unit
PSM	Protocol/Service Multiplexer
RFU	Reserved for Future Use
SDP	Service Discovery Protocol
uint48	unsigned 48-bit integer
URL	Uniform Resource Locator
UTF	Unicode Transformation Format
UUID	universally unique identifier

Table 6.1: Acronyms and abbreviations

7 References

- [1] Bluetooth Core Specification, Version 5.2 or later
- [2] Bluetooth SIG Assigned Numbers, <https://www.bluetooth.com/specifications/assigned-numbers/>
- [3] Object Transfer Service Specification, Version 1.0
- [4] ID3 tag version 2.4.0 – Main Structure (<http://id3.org/id3v2.4.0-structure>)
- [5] GATT Specification Supplement, <https://www.bluetooth.com/specifications/gss/>
- [6] Portable Network Graphics (PNG) Specification (Second Edition) (<https://www.w3.org/TR/2003/REC-PNG-20031110/>)