BLUETOOTH DOC	Date / Year-Month-Day 2007-04-16	Approved Adopted	Revision V13	Document No AVRCP_SPEC
Prepared Audio Video WG	e-mail address avv-feedback@bluetooth.org			N.B.

AUDIO/VIDEO REMOTE CONTROL PROFILE

Abstract

This profile defines the requirements for *Bluetooth*® devices necessary for the support of the Audio/Video Remote Control usage case. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the Audio/Video Remote Control usage case.

Revision History

Revision	Date	Comments
0.5	April 2001	Release to Associates
0.7	June, 2001	Release to Associates
0.9	September, 2001	Release to Associates and Early Adopters
Voting Draft 0.95	October, 2001	Release to Associates and Early Adopters
Voting Draft 0.95 a	February 11, 2002	Release to Associates and Early Adopters, small clarifications based on IOP and feedback.
0.95b	March 2002	Adopted 0.95
Voting Draft 1.00	May 2002	Release for Voting Draft
Voting Draft 1.00 a	February 2003	Release for Voting Draft
Version 1.0	May 2003	Title and header changed
Version 1.1 RC1	August 2003	Updated to support and use Bluetooth Core 1.2
Version 1.1 RC2	August 2003	Chapter 2.4 identifies useful services in BT 1.2
Version 1.1 RC3	October 2003	Contributor list updated
Version 1.1 RC4	October 2003	AV/C reference updated
D13r00	15 Aug 2005	Updated for core release v1.2 or later
D13r01	18 April 2006	Updated with Metadata Transfer FIPD content
D13r02	15 May 2006	Updated SDP record, InformDisplayableCharacterSet comments and specific Metadata Transfer introduction comments from the AV WG members
D13r03	16th May 2006	Updated with Patric's description of Basic Group Navigation. Misc. editorial comments
D13r04	12th June 2006	Updated with Laurent's PDU examples and Section 5.5
D13r05	23rd June 2006	Updated with new timers for metadata transfer and minor edits
D13r06	05 October 2006	Converted into a Voting Draft for Metadata Transfer; included Issues 1894, 1895, 1896, 1897, 1898, 1899, 1900, 1901, 1902, 1903, 1904
D13r07	14th October 2006	Updates during F2F meeting: included issues on interleaving commands and clarified RequestContinue PDU ID usage; included Assigned Numbers dependencies and Reference; repaired the Reference section; some editorial and cosmetic updates; updated contributor list; reformatted and added captions to tables in chapter 5; repaired links within document
D13r08	30th October 2006	Update after comment from BARB: Included reference to AV/C
D13r09	31st October 2006	Update after further comments from BARB: removed last paragraph in section 1.1; corrected spellings
D13r10	28 February 2007	Incorporate errata 2077 and 2079
V13	16 April 2007	Prepare for adoption

Contributors

Name	Company
Alexander Hanke	Audi
Ash Kapur	Broadcom
Rüdiger Mosig	BMS
Gordon Downie	CSR
Souichi Saito	Denso
Morgan Lindqvist	Ericsson
Wim Koster	Ericsson
Rene Kuiken	Ericsson

Masahiko Nakashima	Fujitsu
Baskar Subramanian	Impulsesoft
K.A Srinivasan	Impulsesoft
Ilya Goldberg	Matsushita
Tsuyoshi Okada	Matsushita
Thomas Karlsson	Mecel
Jurgen Schnitzler	Nokia
Kalervo Kontola	Nokia
Martti Niska	Nokia
Thomas Block	Nokia
Vesa Lunden	Nokia
Sebastien Henrio	Parrot
Erik Schylander	Philips
Shaun Barrett	Philips
Christian Bouffioux	Philips
Geert Knapen	Philips
Emmanuel Mellery	Philips
Laurent Meunier	Philips
Scott Walsh	Plantronics
Dmitri Toropov	Siemens
Masakazu Hattori	Sony
Harumi Kawamura	Sony
Rudiger Mosig	Sony
Yoshiyuki Nezu	Sony
Hiroyasu Noguchi	Sony
Tomoko Tanaka	Sony
Atsushi Ichise	Sony
Wilhelm Hagg	Sony
Masahiko Seki	Sony
Dick de Jong	Sony Ericsson
Patric Lind	Sony Ericsson
Siân James	Symbian
Junko Ami	Toshiba
Yoshiaki Takabatake	Toshiba
Ichiro Tomoda	Toshiba
Makoto Kobayashi	Toshiba
Shuichi Sakurai	Toshiba
Makoto Yamashita	Toshiba

Disclaimer and Copyright Notice

The copyright in this specification is owned by the Promoter Members of *Bluetooth*® Special Interest Group (SIG), Inc. ("*Bluetooth* SIG"). Use of these specifications and any related intellectual property (collectively, the "Specification"), is governed by the Promoters Membership Agreement among the Promoter Members and *Bluetooth* SIG (the "Promoters Agreement"), certain membership agreements between *Bluetooth* SIG and its Adopter and Associate Members (the "Membership Agreements") and the *Bluetooth* Specification Early Adopters Agreements (1.2 Early Adopters Agreements) among Early Adopter members of the unincorporated *Bluetooth* SIG and the Promoter Members (the "Early Adopters Agreement"). Certain rights and obligations of the Promoter Members under the Early Adopters Agreements have been assigned to *Bluetooth* SIG by the Promoter Members.

Use of the Specification by anyone who is not a member of *Bluetooth* SIG or a party to an Early Adopters Agreement (each such person or party, a "Member"), is prohibited. The legal rights and obligations of each Member are governed by their applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Any use of the Specification not in compliance with the terms of the applicable Membership Agreement, Early Adopters Agreement or Promoters Agreement is prohibited and any such prohibited use may result in termination of the applicable Membership Agreement or Early Adopters Agreement and other liability permitted by the applicable agreement or by applicable law to *Bluetooth* SIG or any of its members for patent, copyright and/or trademark infringement.

THE SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, SATISFACTORY QUALITY, OR REASONABLE SKILL OR CARE, OR ANY WARRANTY ARISING OUT OF ANY COURSE OF DEALING, USAGE, TRADE PRACTICE, PROPOSAL, SPECIFICATION OR SAMPLE.

Each Member hereby acknowledges that products equipped with the *Bluetooth* technology ("*Bluetooth* products") may be subject to various regulatory controls under the laws and regulations of various governments worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of *Bluetooth* products. Examples of such laws and regulatory controls include, but are not limited to, airline regulatory controls, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their *Bluetooth* Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their *Bluetooth* products related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses**. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.**

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST *BLUETOOTH* SIG AND ITS PROMOTER MEMBERS RELATED TO USE OF THE SPECIFICATION.

Bluetooth SIG reserve the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate.

Document Terminology

The Bluetooth SIG has adopted Section 13.1 of the IEEE Standards Style Manual, which dictates use of the words ``shall'', ``should'', ``may'', and ``can'' in the development of documentation, as follows:

- The word *shall* is used to indicate mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).
- The use of the word *must* is deprecated and shall not be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.
- The use of the word *will* is deprecated and shall not be used when stating mandatory requirements; *will* is only used in statements of fact.
- The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited (*should* equals *is recommended that*).
- The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted*).
- The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

Contents

1 Introduction

1.1 Scope

The Audio/Video Remote Control Profile (AVRCP) defines the features and procedures required in order to ensure interoperability between Bluetooth devices with audio/video control functions in the Audio/Video distribution scenarios. This profile specifies the scope of the AV/C Digital Interface Command Set (AV/C command set, defined by the 1394 Trade Association) to be applied, and it realizes simple implementation and easy operability. This profile adopts the AV/C device model and command format for control messages, and those messages are transported by the Audio/Video Control Transport Protocol (AVCTP).

In this profile, the controller translates the detected user action to the A/V control signal, and then transmits it to a remote Bluetooth device. The functions available for a conventional infrared remote controller can be realized in this profile. In addition to this the profile uses Bluetooth specific extensions to support transfer of metadata related to content to be transferred between Bluetooth devices. The remote control described in this profile is designed specific to A/V control. Other remote control solutions using Bluetooth wireless technology may be applied for general Bluetooth devices including A/V devices.

Note that the Audio/Video Remote Control Profile does not handle the audio/video streaming. Devices that support this profile may support audio/video streaming by also implementing the Advanced Audio Distribution Profile and/or Video Distribution Profile.

1.2 Profile Dependencies

In Figure 1.1, the structure and dependencies of the Audio/Video Remote Control Profile are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it.

As indicated in the figure, the Audio/Video Remote Control Profile is dependent upon the Generic Access Profile. The details regarding the profile are provided in Section 10, Generic Access Profile Requirements.

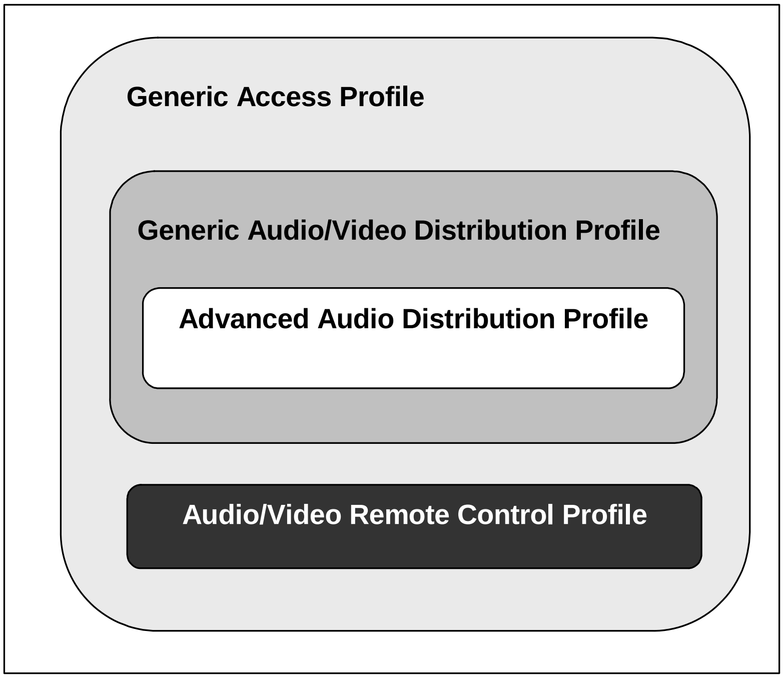

Generic Access Profile
Generic Audio/Video Distribution Profile
Advanced Audio Distribution Profile
Audio/Video Remote Control Profile

Figure 1.1: Audio/Video Remote Control Profile Dependency

1.3 Symbols and Conventions

1.3.1 Requirement Status Symbols

In this document, the following symbols are used:

‘M’ for mandatory to support (used for capabilities that shall be used in the profile).

‘O’ for optional to support (used for capabilities that may be used in the profile).

‘X’ for excluded (used for capabilities that may be supported by the unit but that shall never be used in the profile).

‘C’ for conditional to support (used for capabilities that shall be used in case a certain other capability is supported).

‘N/A’ for not applicable (in the given context it is impossible to use this capability).

Some excluded capabilities are the ones that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade the operation of devices following this profile. Even if such features exist, which can occur when the device supports different profiles, they should never be activated while the device is operating within this profile.

1.3.2 Definition

1.3.2.1 RFA

Reserved for Future Additions. Bits with this designation shall be set to zero. Receivers shall ignore these bits.

1.3.2.2 RFD

Reserved for Future Definition. These bit value combinations or bit values are not allowed in the current specification but may be used in future versions. The receiver shall check that unsupported bit value combination is not used.

1.3.3 Conventions

In this profile, protocol signals are exchanged by initiating procedures in communicating devices and by exchanging messages. Signaling diagrams use the conventions of Figure 1.2: Signaling Conventions. Both A and B represent devices playing specific roles, as defined in Section 2.2, Configuration and Roles. Specific arrow styles are used in the diagrams to indicate the relevant procedures initiated by the participant devices and the exchanged messages.

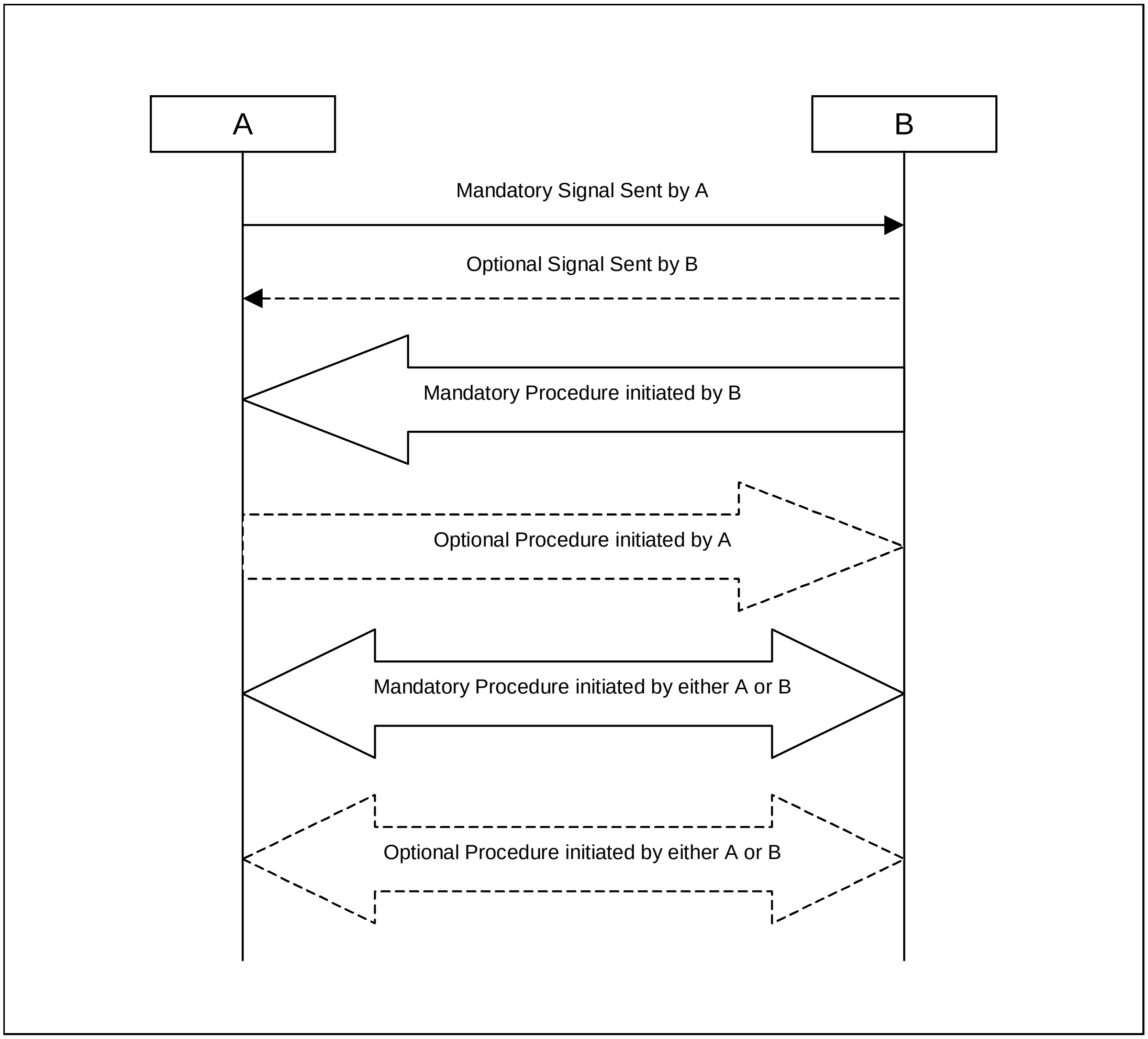

A
B
Mandatory Signal Sent by A
Optional Signal Sent by B
Mandatory Procedure initiated by B
Optional Procedure initiated by A
Mandatory Procedure initiated by either A or B
Optional Procedure initiated by either A or B

Figure 1.2: Signaling Conventions

1.3.4 Notation for Timers

Timer is introduced, specific to this profile. To distinguish them from timers used in the Bluetooth protocol specifications and other profiles, these timers are named in the following format:

- “T_{mmm} (nnn)” for timers, where mmm specifies the different timers used and nnn specifies time in milliseconds.

2 Profile Overview

2.1 Profile Stack

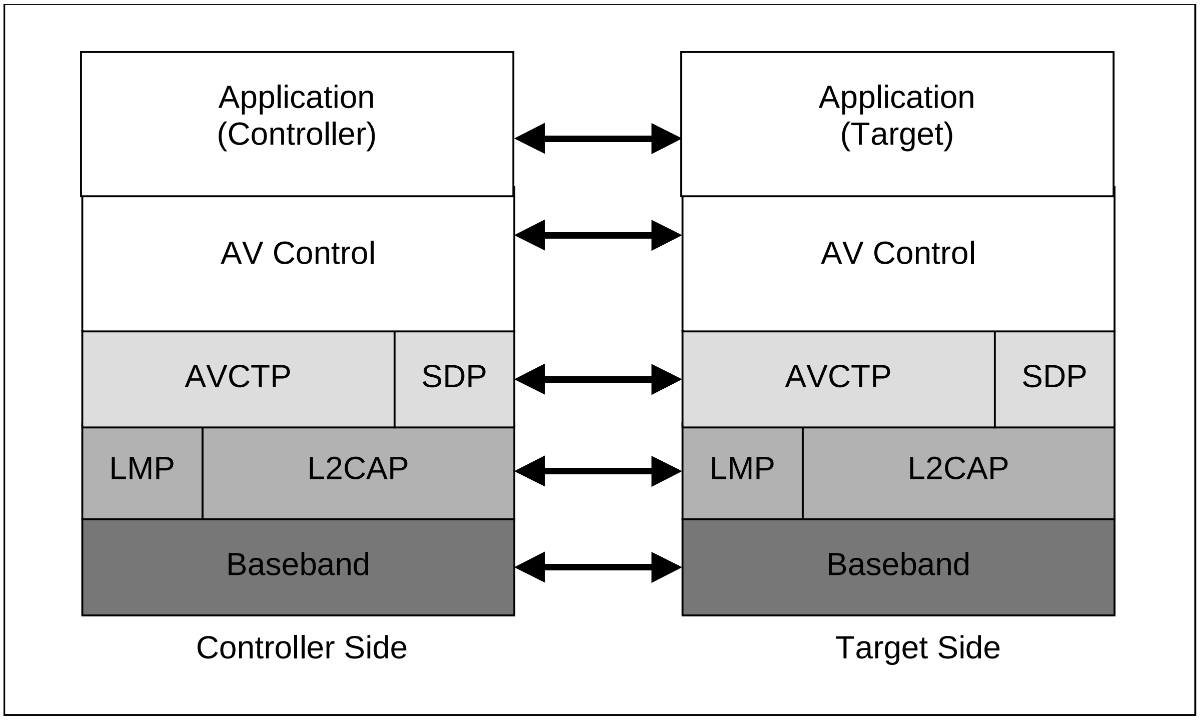


Figure 2.1: Protocol Model

The Baseband, LMP, and L2CAP are the OSI layer 1 and 2 Bluetooth protocols. AVCTP defines the procedures and messages to be exchanged for controlling A/V devices. SDP is the Bluetooth Service Discovery Protocol [10]. AV control is the entity responsible for A/V device control signaling; this signaling is AV/C command-based.

2.2 Configuration and Roles

For the configuration examples for this profile, refer to the figures shown in Section 2.3.

The following roles are defined for devices that comply with this profile:

- The **controller (CT)** is a device that initiates a transaction by sending a command frame to a target. Examples for CT are a personal computer, a PDA, a mobile phone, a remote controller or an AV device (such as headphone, player/recorder, timer, tuner, monitor etc.).
- The **target (TG)** is a device that receives a command frame and accordingly generates a response frame. Examples for TG are an audio player/recorder, a video player/recorder, a TV, a tuner, an amplifier or a headphone.

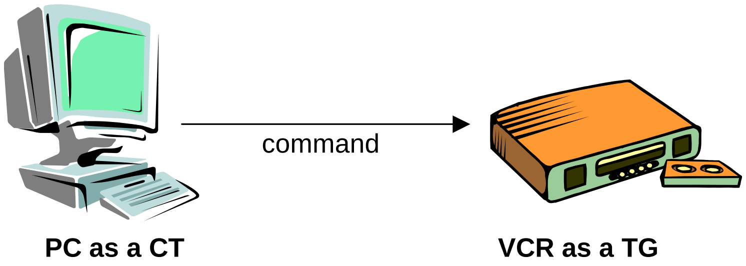


Figure 2.2: Controller and target

2.3 User Requirements

2.3.1 Scenarios

User requirements and scenarios for the configuration examples are described in this section.

The usage model of AVRCP is specific in a way that user action manipulates the control, but there is no limitation to perform the features in audio/video devices. AVRCP is capable to manipulate the menu function that is already commonly used for analogue devices for various features such as adjustment of TV brightness or hue, or VCR timer. With this menu function, AVRCP is designed so that any type features can be supported.

A user can learn the status information of a device using display on the body such as LED or LCD, as well as OSD (On Screen Display) method.

2.3.1.1 Remote Control from Separate Controller

In the configuration shown in Figure 2.3 below, the remote controller is the CT of the transaction. Command frames from the remote controller are sent to the portable disc player as a TG. An audio stream is sent from the portable disc player to the headphone. The headphone simply receives the audio stream and is not involved in the transaction between the remote controller and the portable disc player. A trigger of the transaction is made by a user from the remote controller, when he/she wishes to control the portable disc player.

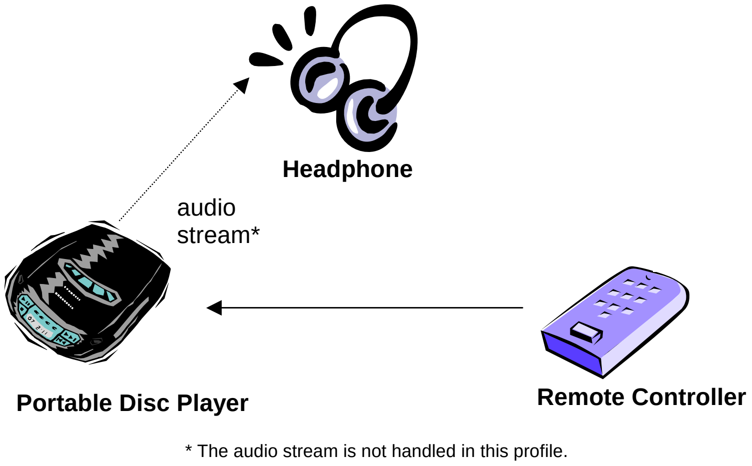


Figure 2.3: Remote Control from Separate Controller

2.3.1.2 Remote Control and Audio Stream between Two Devices

In the configuration shown in Figure 2.4 below, the CT is the headphone and the portable disc player is the TG.

A trigger of the transaction is made by a user from the remote controller that accompanies the headphone, when he/she wishes to control the portable disc player.

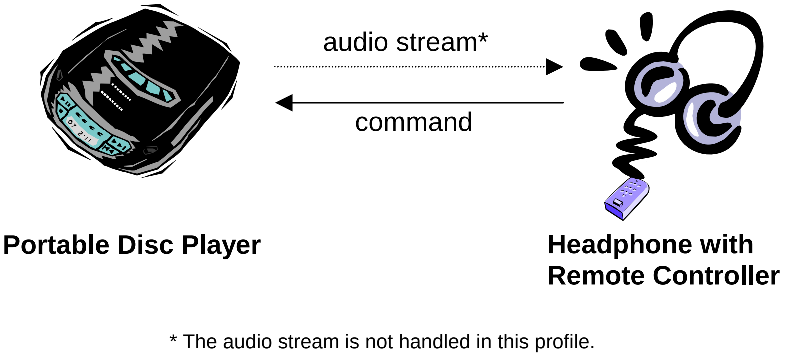
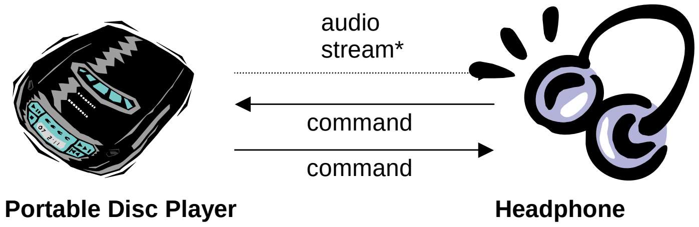

audio stream*
command
Portable Disc Player
Headphone with Remote Controller
* The audio stream is not handled in this profile.

Figure 2.4: Remote Control and Audio Stream between Two Devices

2.3.1.3 Mutual Remote Control within a Piconet

In the configuration shown in Figure 2.5 below, both the headphone and the portable disc player are capable of working as remote controllers.

For example, the portable disc player becomes a CT if it controls the volume of the headphone that becomes a TG. On the other hand, the headphone becomes a CT when it sends a command to start playback or stop playing to the portable disc as a TG.

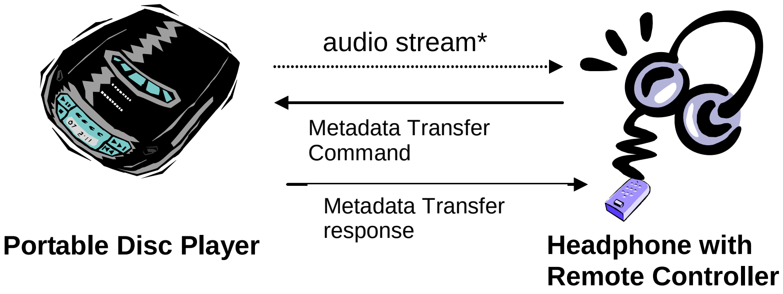


* The audio stream is not handled in this profile.

Figure 2.5: Mutual Remote Control within a Piconet

2.3.1.4 Remote controller with LCD

In the configuration shown in Figure 2.6 below, the headphone with a LCD remote controller is a CT. It receives media metadata information by sending commands to the media player which is the TG. The remote controller <u>can</u> have an LCD to present received data to a user.



* The audio stream is not handled in this profile.

Figure 2.6: Headphone with LCD connected to media player

2.3.2 User Expectations

In this section, user expectations and related restrictions of AVRCP are described.

Although a device <u>may</u> implement only AVRCP as shown in 2.3.1.1, it is assumed that, in most cases, an A/V distribution profile co-exists in a device. Items described in this section <u>shall</u> be considered according to the condition; whether only AVRCP is implemented, or one or more AV distribution profiles co-exist in the device.

2.3.2.1 Configuration

AVRCP is based on the control over point to point connection within a piconet. For this profile, it is assumed that the use case is active between the two devices. Note that one or more CTs <u>may</u> exist within a piconet. (Refer to 2.3.1.3)

A controller <u>may</u> support several targets, and the detail of control such as target selection is not defined in AVRCP.

2.3.2.2 Limited Latency

The responsiveness of remote control operations is an important feature of AVRCP. It is expected that the system reacts in a timely manner in order to avoid uncontrollable situations like system overload by repeated commands.

Latency figures depend on application. Additional information on the desired delay is provided in 16 Appendix A (Informative): Example of Latency.

CT and TG interoperate through L2CAP channel connections. In case the TG is a master, it is required to poll the slaves on a regular basis in order to satisfy the application QoS requirements. It is recommended that the polling rate is approximately 10 Hz.

2.3.2.3 Power Management

The discussions below are intended to be for application information only: there are no mandatory usages of the low power modes for AVRCP.

It is assumed that battery powered devices are common in the usage model of AVRCP, in case that CT is a handheld device. The device is recommended to ensure comparable service grade to the existing infrared product range.

Duplex radio systems suffer from higher power consumption compared to the simple infrared transmission controller. To compensate this fundamental drawback, dynamic use of low power modes is recommended especially when only AVRCP is implemented in a device.

Regarding the details of the low power modes, refer to the Specification of the Bluetooth System, Core, Baseband [7] and Link Manager Protocol [9]. Appropriate low power mode strategy partly depends on applications.

2.3.2.4 User Action

The user action or media status change is required in most cases in AVRCP. Applications shall be designed based on this characteristic. It is possible to design simple automatic operation without a user action; such as a timer function that sends a command to start recording at pre-set time, within this profile.

2.4 Profile Fundamentals

The profile fundamentals, with which all applications shall comply, are the followings.

1. Use of security features in link level such as authorization, authentication and encryption are optional. Support for authentication and encryption is mandatory, such that the device can take part in the corresponding procedures if requested from a peer device.
2. A link shall be established before commands can be initiated or received.
3. There are no fixed master/slave roles.
4. In this profile, the A/V functions are classified into four categories defined in Section 4.8. All devices that conform to this profile shall support at least one category, and may support several categories.

2.5 Conformance

When conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process mandatory). This also applies to optional and conditional capabilities, for which support is indicated, and subject to verification as part of the Bluetooth certification program.

3 Application Layer

This section describes the feature requirements on units complying with the Audio/Video Remote Control Profile.

3.1 Feature Support

The table below shows the features requirements for this profile. Note that a device may have both CT and TG capabilities. In that case, features for both CT and TG are required.

	Feature	**Support in CT**	**Support in TG**
1.	Connection establishment for control	M	O
2.	Release connection for control	M	M
3.	Sending UNIT INFO command	O	X
4.	Receiving UNIT INFO command	X	M
5.	Sending SUBUNIT INFO command	O	X
6.	Receiving SUBUNIT INFO command	X	M
7.	Sending VENDOR DEPENDENT command	C3	X
8.	Receiving VENDOR DEPENDENT command	X	C3
9.	Sending PASS THROUGH command	M	X
10.	Receiving PASS THROUGH command	X	M
11.	Capabilities	O	C1
12.	Player Application Settings	O	O
13.	Metadata Attributes	O	C1
14.	Notifications	C2	C2
15.	Continuation	C2	C2
16.	Basic Group Navigation	O	O

Table 3.1: Application Layer Features

C1 – Mandatory if Target supports Category 1 or optional otherwise
C2 – Mandatory if Controller supports Metadata Attributes or optional otherwise
C3 – Mandatory if any of 11, 12, 13, 14, 15 features are supported or optional otherwise

Features 3.1-11 to 3.1-16 shall be as a whole termed as Metadata transfer feature in this document.

3.2 Feature Mapping

The table below maps each feature to the procedures used for that feature. All procedures are mandatory if the feature is supported.

1.	Connection establishment	Connection for control	4.1.1
2.	Connection release	Release connection for control	4.1.2
3.	Sending UNIT INFO command	Procedure of AV/C command	4.1.3

4.	Receiving UNIT INFO command	Procedure of AV/C command	4.1.3
5.	Sending SUBUNIT INFO command	Procedure of AV/C command	4.1.3
6.	Receiving SUBUNIT INFO command	Procedure of AV/C command	4.1.3
7.	Sending VENDOR DEPENDENT command	Procedure of AV/C command	4.1.3
8.	Receiving VENDOR DEPENDENT command	Procedure of AV/C command	4.1.3
9.	Sending PASS THROUGH command	Procedure of AV/C command	4.1.3
10.	Receiving PASS THROUGH command	Procedure of AV/C command	4.1.3
11.	Capabilities	Procedure of Metadata Transfer	4.7.3
12.	Player Application Settings	Procedure of Metadata Transfer	4.7.4
13.	Metadata Attributes	Procedure of Metadata Transfer	4.7.5
14.	Notifications	Procedure of Metadata Transfer	4.7.6
15.	Continuation	Procedure of Metadata Transfer	4.7.7
16.	Basic Group Navigation	Procedure of Metadata Transfer	4.7.8

Table 3.2: Application Layer Feature to Procedure Mapping

The general procedure of Metadata Transfer is described in section 4.1.5.

4 Control Interoperability Requirements

The interoperability requirements for an entity that is compatible with the AVRCP are completely contained in this chapter. The requirements directly relate to the application layer features.

4.1 Procedure

4.1.1 Connection for Control

An L2CAP connection establishment for AVCTP may be initiated by the CT or by the TG. An internal event or an event generated by a user, such as turning the power on, initiates the connection establishment.

Note: Only one L2CAP connection shall be established between AVCTP entities. If the connection already exists, the CT/TG shall not initiate the connection request.

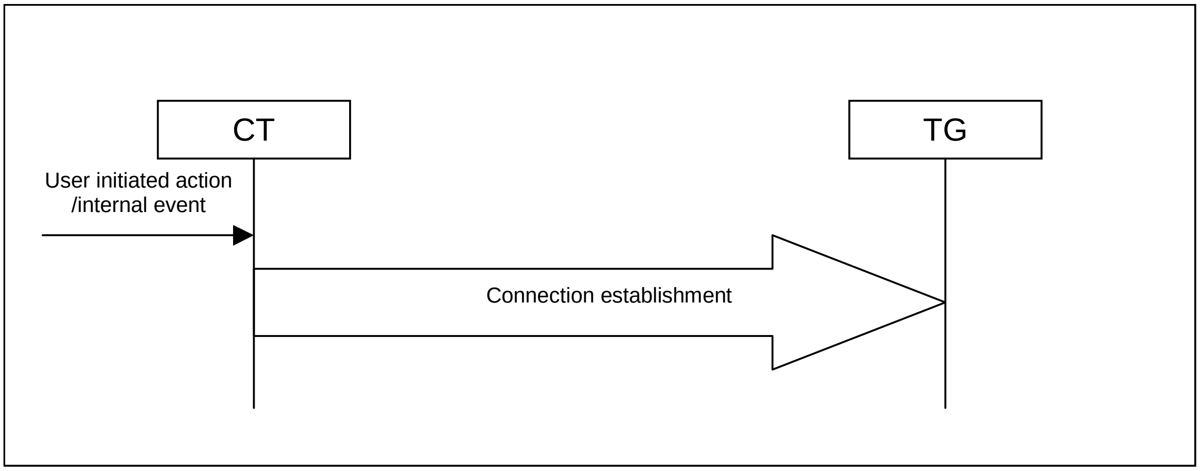


Figure 4.1: Connection Establishment Initiated by CT

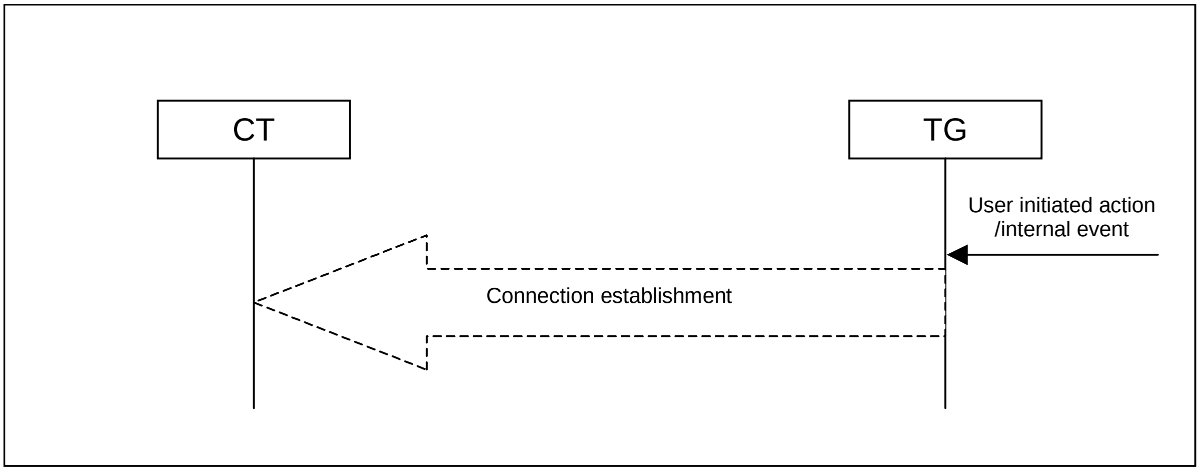


Figure 4.2: Connection Establishment Initiated by TG

4.1.2 Release Connection for Control

Release of an L2CAP connection for AVCTP may be initiated by the CT or by the TG. An internal event or an event generated by a user, such as turning the power off, initiates the connection release.

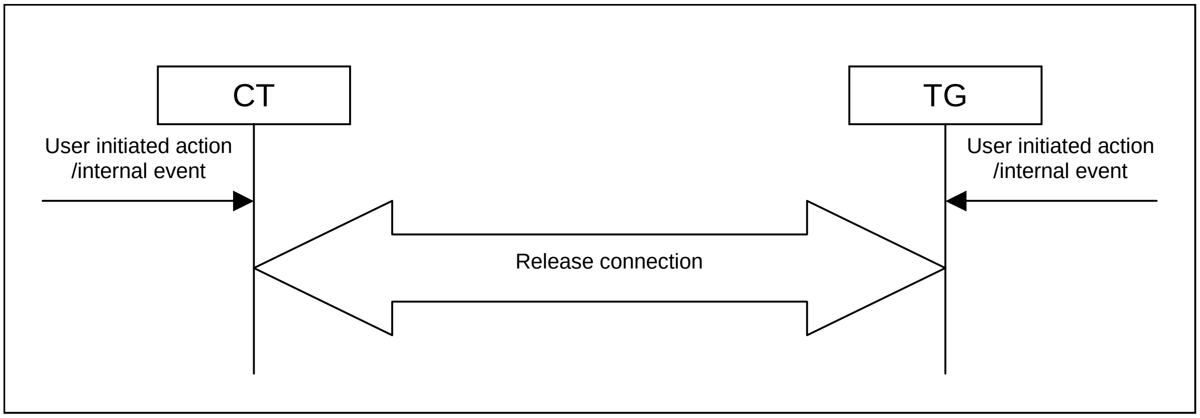

CT
TG
User initiated action /internal event
User initiated action /internal event
Release connection

Figure 4.3: Connection Release Initiated by CT or TG

4.1.3 Procedure of AV/C Command

Upon an internal or an event generated by a user, the CT shall initiate connection establishment if a connection has not been established by then. Once the connection is established, it is able to send an AV/C command.

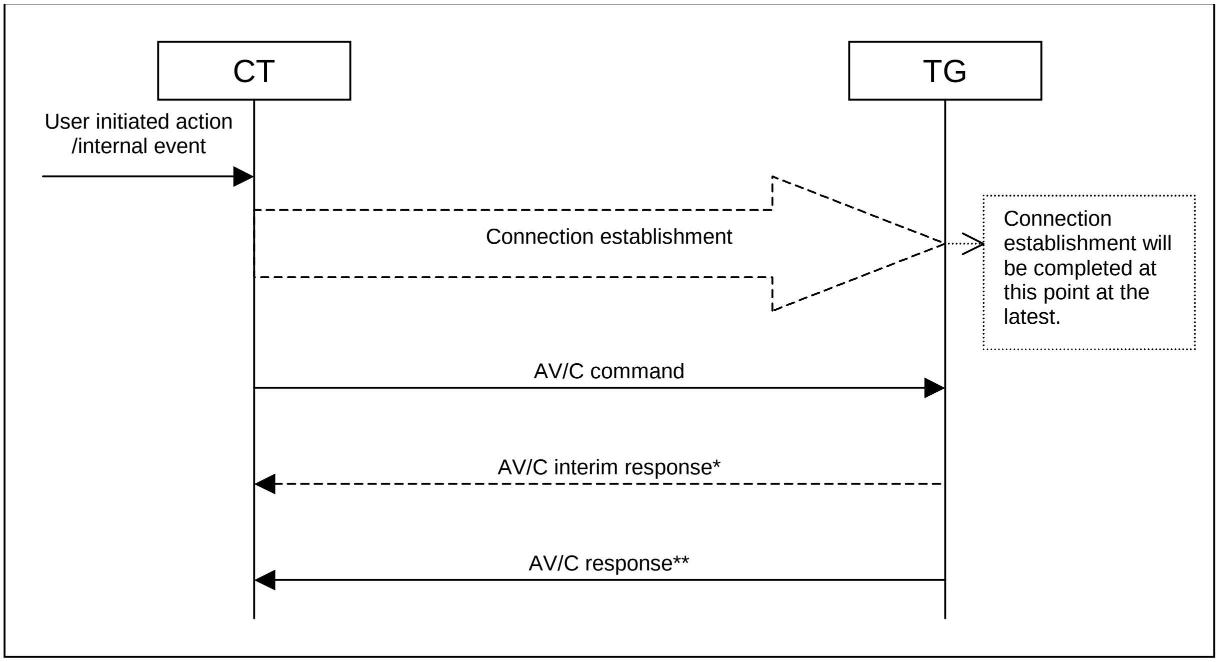

CT
TG
User initiated action /internal event
Connection establishment
Connection establishment will be completed at this point at the latest.
AV/C command
AV/C interim response*
AV/C response**

Figure 4.4: Procedure of AV/C Command

*: AV/C interim response may be returned in response to a VENDOR DEPENDENT command. AV/C interim response shall not be returned for other commands.

**: In some exceptional cases, the TG may not return a response. For details, refer to the AV/C General Specification.

The following table shows the list of possible AV/C commands to be exchanged in this profile:

	Command	CT	TG
1.	UNIT INFO	O	M
2.	SUBUNIT INFO	O	M
3.	VENDOR DEPENDENT	C	C
4.	PASS THROUGH	M	M

Table 4.1: List of Possible AV/C Commands

C - Mandatory if any of 3.1-11 to 3.1-15 is supported optional otherwise
Requirements for CT refer to the ability to send a command.
Requirements for TG refer to the ability to respond to a command.

4.1.4 AV/C Command Operation

This section describes the operation procedure of AV/C command exchange shown in Figure 4.4 with example. For more information of the AV/C unit/subunit model and AV/C command operation [1], refer to AV/C General Specification [1] and AV/C Panel Subunit Specification [2].

The AV/C General Specification covers the AV/C general command and response model, unit/subunit model, and standard unit and subunit commands. An AV/C subunit is an instantiation of a logical entity that is identified within an AV/C unit. An AV/C subunit has a set of coherent functions that the electronic device provides. Functions are defined for each category of devices in its subunit specification. (Monitor, Audio, Tape recorder/player, Disc, Tuner, etc.).

The AV/C command set consists of the AV/C General Specification and each subunit command. In the AV/C General Specification, the UNIT INFO command and SUBUNIT INFO command are both mandatory. For subunit commands, the mandatory commands are defined in each subunit specification, and it depends on the device implementation which subunit to support.

The UNIT INFO command is used to obtain information that pertains to the AV/C unit as a whole. The response frame includes information of the vendor ID of the TG and subunit type that best describes the unit. The information of vendor ID may be used to investigate the vendor of TG before using VENDOR DEPENDENT command. For example of subunit type, a VCR device may return the *unit_type* of the tape recorder/player, even though the VCR has a tuner. In this profile, the panel subunit is the main function. It is also possible that other subunits may be returned if other profiles co-exist in the device.

The SUBUNIT INFO command is used to obtain information about the subunit(s) of an AV/C unit. A device with this profile may support other subunits than the panel subunit if other profiles co-exist in the device, which can be found with the SUBUNIT INFO command. With this command, a typical AV/C controller manipulates AV/C function discovery.

The VENDOR DEPENDENT command permits module vendors to specify their own set of commands and responses for AV/C units or subunits determined by the AV/C address that is contained in the AV/C frame. The vendor dependent commands are used by this specification. Please refer to 4.1.5.

The main feature of this profile is the remote control performed by the PASS THROUGH command of the Panel subunit. The Panel subunit provides a user-centric model for actuating the controls on a device. The controller controls the Panel subunit according to the user operation using certain controller-dependent manners. The user manipulates the user interface on the display or operates a button, and then the controller sends commands to the panel subunit. In response to these commands, the Panel subunit performs some action(s). Even though there may be several subunits in a TG, the TG shall have only one panel subunit. Unlike many other AV/C subunits, the panel subunit does not directly deal with media streams itself. The main purpose for using a panel subunit is to allow it to translate the incoming user action commands into internal actions, which affect other subunits and/or the unit, and dispatch them to an appropriate subunit or unit inside the TG using the TG-dependent manner. The result of these actions may have an effect on media streams. This profile uses the PASS THROUGH command, which is one of the subunit commands defined in the Panel Subunit Specification. A controller conveys a user operation to a TG by the PASS THROUGH command.

4.1.5 Procedure of Metadata Transfer

The Procedure of Metadata Transfer is an extension of the AV/C Digital Interface Command Set General Specification specified by 1394 Trade Association. It enables more sophisticated control functionality as well as the handling of metadata such as song and artist information.

The extension is implemented by defining VENDOR DEPENDENT and PASS THROUGH commands within the framework of the 1394 specifications. A vendor ID for the Bluetooth SIG is used to differentiate from real vendor specific commands.

4.2 AVCTP Interoperability Requirements

4.2.1 Transaction Labels

On the CT side, it is application-dependent how transaction labels are handled, and therefore it is not defined in this specification. On the TG side, the transaction label received in an AVCTP command frame shall be used as the transaction label returned in the possible corresponding AVCTP response frame. In case several response frames are sent as reaction to one AVCTP command, all response frames shall use the same value of transaction label in the received command frame.

4.2.2 Message Fragmentation

The support of AVCTP packet fragmentation in this profile is as follows:

Procedure of AV/C Command	**Non-Fragmented AVCTP Message**		**Fragmented AVCTP Message**	
	Support in CT	Support in TG	Support in CT	Support in TG
UNIT INFO	M	M	X	X
SUBUNIT INFO	M	M	X	X
VENDOR DEPENDENT	M	M	C1	C2
PASS THROUGH	M	M	C1	C2

Table 4.2: AVCTP Fragmentation

C1, C2: In case a vendor defines a VENDOR DEPENDENT command or a vendor unique operation_id of a PASS THROUGH command that is longer than the L2CAP MTU, and if a device implements one, it is M (mandatory) to support the fragmented AVCTP message. If not, it is X (excluded). All metadata transfer commands use VENDOR DEPENDENT command and so support for fragmentation is mandatory for metadata transfer commands (refer to Figure 3-1).

4.2.3 Profile Identifier of AVCTP Message Information

Refer to Bluetooth Assigned Numbers [6] for the value of the profile Identifier for this profile.

Note: The value of Service Class for CT is “A/V Remote Control”, while the value for TG is “A/V Remote Control Target”. The value of Profile Identifier is the same for CT and TG, which is ”A/V Remote Control”.

4.3 AV/C Command and Response

AV/C command and response frames are encapsulated within the AVCTP Command/Response Message Information field, as described in AVCTP [3].

4.3.1 AV/C Transaction Rules

An AV/C transaction consists of one message containing a command frame addressed to the TG and zero or more messages containing a response frame returned to the CT by the TG. The TG is required to generate a response frame within specified time periods.

All transactions except Metadata Transfer VENDOR DEPENDENT commands shall comply with the following time period. TG shall respond to any command within a time period of T_{RCP} (100) counting from the moment a command frame is received.

For some transactions, the TG may not be able to complete the request or determine whether it is possible to complete the request within the T_{RCP} (100) allowed. In this case, the TG shall return an initial response code in INTERIM with the expectation that the final response follow later.

Specifically for Metadata Transfer VENDOR DEPENDENT commands the following time periods are defined.

T_{MTC} (200) is the time period before which TG is expected to generate a response frame for CONTROL commands.

T_{MTP} (1000) is the time period before which TG is expected to generate a response frame for interim response for NOTIFY commands and final response for STATUS commands.

Note: INTERIM response may be returned in response to other VENDOR DEPENDENT command. INTERIM response shall not be returned for any other commands.

For more detail regulations, refer to the AV/C General Specification [1].

4.3.2 AV/C Command Frame

An AV/C command frame contains up to 512 bytes of data, and it is contained in the AVCTP Command/Response Message Information field. An AV/C command frame has the structure shown below.

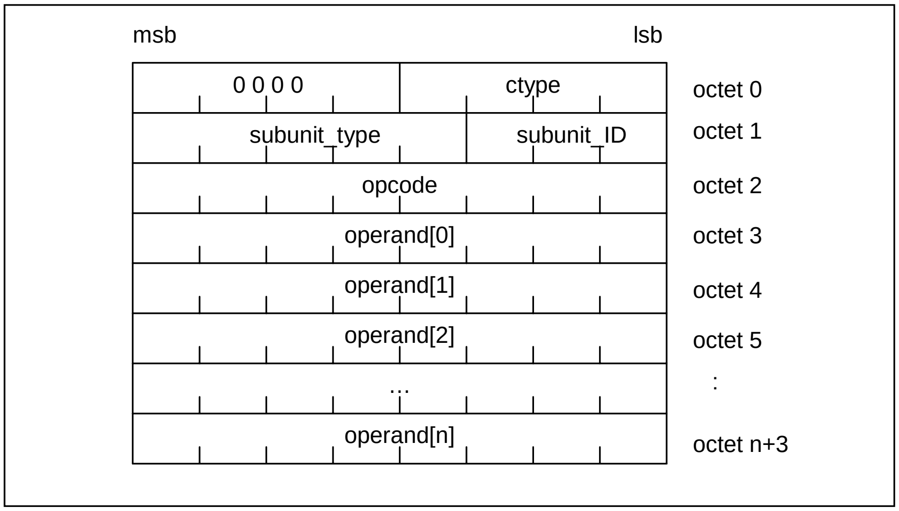


Figure 4.5: AV/C Command Frame

All of the operands are optional and are defined based on the values of *ctype, subunit_type,* and *opcode.*

4.3.3 AV/C Response Frame

An AV/C response frame is contained in the AVCTP Command/Response Message Information field, and it has the structure shown in the figure below.

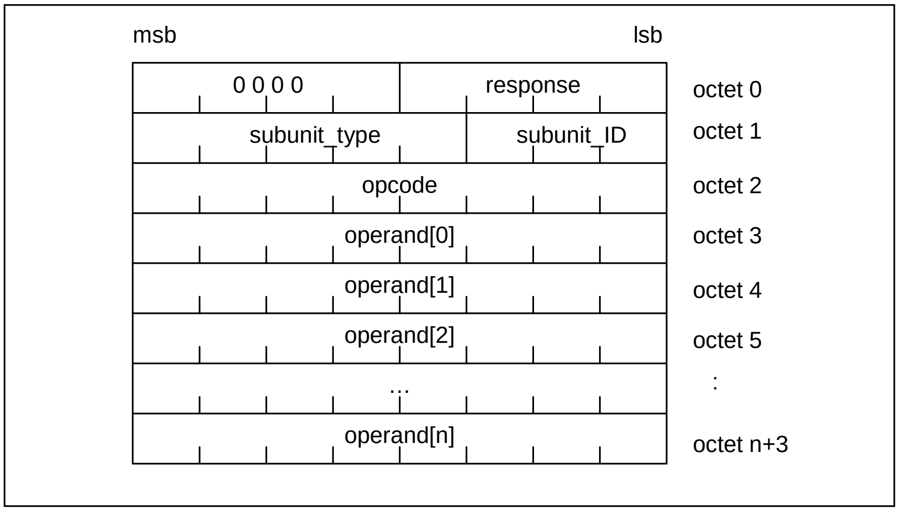


Figure 4.6: AV/C Response Frame

All of the operands are optional and are defined based on the values of *ctype, subunit_type,* and *opcode.*

4.3.4 AV/C Frame Fields

For the fields and code values for AV/C command and response frames listed below, as well as the definition of reserved field and reserved value, refer to the AV/C General Specification.

- Command type codes (*ctype)*
- Response codes (*response*)
- AV/C address (*subunit_type, subunit_ID*)
- Subunit_type and subunit_ID encoding
- Operation (*opcode*)
- Operands

4.4 Supported Unit Commands

The unit commands shown in the following table are used in this profile. For unit commands, the *AV/C address* field of AV/C command frame shall indicate the value for unit.

Opcode	**Support in CT**			**Support in TG**			**Comments**
	CONTROL	STATUS	NOTIFY	CONTROL	STATUS	NOTIFY	
UNIT INFO	N/A	O	N/A	N/A	M*	N/A	Reports unit information
SUBUNIT INFO	N/A	O	N/A	N/A	M*	N/A	Reports subunit information

Table 4.3: Supported Unit Commands

*: These commands shall be supported in AV/C-compliant devices to maintain the compatibility with the existing AV/C implementations.

4.4.1 UNIT INFO Command

As defined in the AV/C General Specification, the UNIT INFO status command is used to obtain information that pertains to the unit as a whole. For details of the UNIT INFO command, refer to the AV/C General Specification [1].

In the *unit_type* field of a response frame, a code for a subunit type that represents the main function of the unit shall be shown. If the unit implements only this profile, it shall return the PANEL subunit in the response frame.

In the *company_ID* field of a UNIT INFO response frame, the 24-bit unique ID obtained from the IEEE Registration Authority Committee shall be inserted. If the vendor of a TG device does not have the unique ID above, the value 0xFFFFFF may be used.

4.4.2 SUBUNIT INFO Command

As defined in the AV/C General Specification, the SUBUNIT INFO status command is used to obtain information about the subunit(s) of a unit. For details of the SUBUNIT INFO command, refer to the AV/C General Specification [1].

If the unit implements this profile, it shall return PANEL subunit in the *subunit_type* field, and value 0 in the *max_subunit_ID* field in the response frame.

4.5 Supported Common Unit and Subunit Commands

The common unit and subunit commands shown in the following table are used in this profile. For the common unit and subunit command, the *AV/C address* field of the AV/C command frame shall indicate the value for unit or Panel Subunit if the command is one defined in this profile.

4.5.1 VENDOR DEPENDENT Command

The formats of a command frame or a response frame, as well as the compliant usage rules, are as defined in the AV/C General Specification [1].

Opcode	Support in CT			Support in TG			Comments
	CONTROL	STATUS	NOTIFY	CONTROL	STATUS	NOTIFY	
VENDOR DEPENDENT	C	C	C	C	C	C	Vendor-dependent commands

Table 4.4: Vendor Dependent Commands

C: M if Metadata Transfer, O otherwise

For metadata transfer feature support, a predefined VENDOR DEPENDENT command is used. The *company_ID* field of the VENDOR DEPENDENT command shall contain a 24-bit unique ID [0x001958]. This unique *Company_ID* field shall be used by all metadata transfer feature supported PDUs. It is assumed that devices that do not support this metadata transfer related features shall return a response of NOT IMPLEMENTED as per AV/C protocol specification [1].

For metadata transfer feature specific VENDOR DEPENDENT command support, refer to 4.7.

The VENDOR DEPENDENT command other than that defined for metadata transfer feature support shall not be used instead of commands specified in the AVRCP that have the same functionality.

4.6 Supported Subunit Command

The PASS THROUGH command of the Panel subunit is used in this profile. The *operation_id*'s to be used in this profile depend on which A/V function category the device supports. The details of categories are described in Section 4.8 Categories.

For the PASS THROUGH command, the *AV/C address* field of the AV/C command frame shall indicate the value for Panel Subunit.

Opcode	Support in CT			Support in TG			Comments
	CONTROL	STATUS	NOTIFY	CONTROL	STATUS	NOTIFY	
PASS THROUGH	M*	N/A	N/A	M*	N/A	N/A	Used to transfer user operation information from CT to Panel subunit of TG.

Table 4.5: PASS THROUGH Command

M*: Mandatory to support the opcode for PASS THROUGH command. See 0 for support levels of each *operation_id*'s

4.6.1 PASS THROUGH Command

As defined in the AV/C Panel Subunit Specification [2], the PASS THROUGH command is used to transfer user operation information from a CT to Panel subunit on TG. For the details of the PASS THROUGH command, refer to the AV/C Panel Subunit Specification [2].

Metadata transfer feature supports pre-defined PASS THROUGH commands to handle group navigation capability. Refer to § 4.6.1.

4.7 Metadata Transfer Data Representation

All the command and response packets are sent using AV/C Digital Interface Command Set General Specification as specified by 1394 Trade Association. All the packets are exchanged as part of VENDOR DEPENDENT commands defined in the 1394 specifications.

The format for *vendor_dependent_data* field shall be as defined in AV/C specification document [1].

4.7.1 Transfer Byte Order

Packets shall transfer multiple-byte fields in standard network byte order (Big Endian), with more significant (high-order) bytes being transferred before less-significant (low-order) bytes.

4.7.2 Protocol Data Unit Format

The vendor_dependent_data field in the vendor dependent command/response frames will be a Metadata Transfer PDU (Protocol Data Unit).

Every PDU consists of a PDU Identifier, length of all parameters (excluding the parameter length field) and the PDU-specific parameters.

Oct	MSB (7)	6	5	4	3	2	1	LSB (0)
0	PDU ID							
1	Reserved						Packet Type	
2-3	Parameter Length							
4-n	Parameter (Number of parameters determined by Parameter Length)							

Table 4.6: Metadata Transfer PDU format

The PDU fields are briefly described below:

PDU ID: The PDU ID is used to identify the specific command/response with unique identifier for each operation.

Packet Type: The Packet Type field qualifies each packet as either start (Packet Type=01), continue (Packet Type=10), or end packet (Packet Type=11). In the case of a non-fragmented message, this field (Packet Type=00) simply indicates that the message is sent in a single AV/C frame.

The packets are fragmented by the sender so as to be able to accommodate into the 512 bytes AV/C packet size restriction. Receivers have the flexibility to request continuation packets at their convenience from the sender or abort the continuation request. Note: if the L2CAP MTU is less than 512 bytes, AVCTP will also apply fragmentation to each of the AV/C packets. All response fragments shall have the same PDU ID as the original request.

A sender shall not interleave fragmented PDUs. Once a sender has sent a start fragment it shall only send further fragments of that PDU until that PDU is completed or aborted. If a receiver receives a start fragment or non-fragmented Metadata-Transfer message when it already has an incomplete fragment from that sender then the receiver shall consider the first PDU aborted. A PASSTHROUGH command may be interleaved in fragmented Metadata-Transfer communication without aborting it.

Parameter Length: The parameter length field specifies the length of all the parameters following the Parameter Length field in Figure 4.6. In the case of fragmented packets, all packets shall contain the Parameter Length field.

Parameter1 …n: These are the parameters for the specific operations performed and are described in sections below.

An example Metadata Transfer command (GetCapabilities) PDU will be as follows:

Oct	MSB (7)	6	5	4	3	2	1	LSB (0)
0	0x0				Ctype: 0x1 (STATUS)			
1	Subunit_type:0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x0 (VENDOR DEPENDENT)							
3 -5	Company ID: 0x001958, BT SIG registered CompanyID							
6	PDU ID (0x10 - Get Capabilities)							
7	Reserved (0x00)						Packet Type (0x0)	
8 – 9	Parameter Length (0x0001)							
10	Capability ID (0x1)							

Table 4.7: Metadata Transfer Command

The grayed portion in table above indicates the Metadata Transfer PDU inside an AV/C Vendor dependent command frame.

Metadata Transfer Commands

This section discusses the details of the features of metadata transfer support.

4.7.3 Capabilities

CT shall have the ability to query the capabilities of TG. The following capabilities can be queried,

1. List of Company IDs supported by TG
2. List of Event IDs supported by TG. Refer to Appendix-H for Event IDs defined in the specification.

4.7.4 Target player application settings

Player application settings commands provide a mechanism for CT devices to query player application setting attributes on the TG and to get and set specific setting value for these attributes.

All player application settings are available on the target as an <attribute, value> pair. For each player application attribute there shall be multiple possible values, with one of them being the current set value.

The specification defines pre-defined attributes and values for some of the commonly used player application settings, defined in Appendix-F of this specification.

The PDUs allow for extensions to the pre-defined attributes and values defined in the target and are accessible to the controller, along with displayable text. This will allow controllers without the semantic understanding of the target’s player application setting to be able to extend their menu by displaying setting related text and provide users with a mechanism to operate on the player application settings.

Each player application setting has a unique AttributeID and the attributes have values that have a ValueID. Target-defined attributes and values have displayable text associated with them for allowing CT to be able to provide menu extensions to existing media players.

Refer to section 5.2 for the list of PDUs.

4.7.5 Media track metadata attributes transfer

Following capabilities shall be supported,

CT has the capability to access metadata attributes for the currently selected media track.

4.7.6 Event notifications from target device

The following capabilities are provided as part of event notifications from the target device,

1. CT has the capability to access current play status in addition to media track duration and current position of the track.
2. Events that can be monitored on the target are,
    a) Play status events of the current media track,
        - Playing
        - Paused
        - Stopped
        - Seek Forward
        - Seek Rewind
        - Playback position change

    b) Track change events,
        - Change of track
        - Start of track
        - End of track

    c) Device unplugged event, to support target-end external adapters to media devices.

    d) All player application attributes can be registered as events by the CT. The TG shall notify the CT on change in value of the corresponding device setting by the local TG device. .

3. CT devices has capability to provide a NOTIFY AV/C command to the TG, to register for specific events on the TG.
4. The TG shall for every NOTIFY AV/C command send an INTERIM response to the controller with the current status of the registered event within TMTP from the time of registration.
5. On the occurrence of a registered event, TG device shall send a NOTIFY_RESPONSE to the controller with the current status
6. As per AV/C protocol a NOTIFY command terminates after providing a corresponding CHANGED. To comply with this, it is expected that controller devices that need periodic updates on selected events, re-register for those events after the receipt of the corresponding CHANGED.
7. As per AVCTP, there shall be only 16 outstanding transaction labels at any instant of time in a session. This shall limit the number of events that can be simultaneously registered or pending response to 15.

4.7.7 Continuation

Continuation commands provide protocol capability for sender and receiver to be able to segment and reassemble packets over AV/C. The commands include,

Request for continuation packets

Abort continuation of current message.

Packet type on the PDU response from TG shall indicate whether the PDU is a start packet with additional packets available for CT as response to its PDU command. CT shall then request for continuation packets using the Continuation PDU till end of packet is signaled on the PDU packet type.

CT has the option to abort the current PDU continuation response packets by sending the continuation abort PDU anytime after the reception of the first PDU response for the corresponding PDU command.

4.7.8 Group navigation

Group navigation provides the ability for CT to logically view TG media storage as a flat structure. Groups are logical blocks of media tracks defined by TG. This could be play lists, folders or any other logical structure as deemed fit by TG. By doing this CT shall be able to move to next and previous groups on TG without any knowledge on the media storage mechanism of TG.

4.7.9 Metadata Transfer PDUs

The following PDUs are added that shall be used with VENDOR DEPENDENT command:

PDU ID	PDU Name	AV/C Command Type	CT	TG	TG Response Time	Section
	Capabilities					**5.1**
0x10	GetCapabilities	STATUS	M	M	T_{MTP}	5.1.1
	Player Application Settings					**5.2**
0x11	ListPlayerApplicationSettingAttributes	STATUS	M	M	T_{MTP}	5.2.1
0x12	ListPlayerApplicationSettingValues	STATUS	O	M	T_{MTP}	5.2.2
0x13	GetCurrentPlayerApplicationSettingValue	STATUS	C2	M	T_{MTP}	5.2.3
0x14	SetPlayerApplicationSettingValue	CONTROL	C2	M	T_{MTC}	5.2.4
0x15	GetPlayerApplicationSettingAttributeText	STATUS	O	C1	T_{MTP}	5.2.5
0x16	GetPlayerApplicationSettingValueText	STATUS	O	C1	T_{MTP}	5.2.6
0x17	InformDisplayableCharacterSet	CONTROL	O	O	T_{MTC}	5.2.7
0x18	InformBatteryStatusOfCT	CONTROL	O	O	T_{MTC}	5.2.8
	Metadata Attributes					**5.3**
0x20	GetElementAttributes	STATUS	M	M	T_{MTP}	5.3.1
	Notifications					**5.4**
0x30	GetPlayStatus	STATUS	O	M	T_{MTP}	5.4.1
0x31	RegisterNotification	NOTIFY	M	M	T_{MTP}	5.4.2
	Continuation					5.5
0x40	RequestContinuingResponse	CONTROL	M	M	T_{MTC}	5.5.1
0x41	AbortContinuingResponse	CONTROL	M	M	T_{MTC}	5.5.2

Table 4.8: Operations used with VENDOR DEPENDENT command

C1 If player application setting attribute IDs for menu extension (Refer to Appendix F) are supported then mandatory or optional otherwise.

C2 Either Get or Set player application settings shall be mandatory

Requirements for CT refer to the ability to send a command.

Requirements for TG refer to the ability to respond to a command. The AV/C command type of the response PDU shall be per the AV/C specification’s definitions for responses.

For error response PDU the response parameter is always the error code independent of the response format defined for ACCEPTED PDU response for the corresponding PDU command.

All strings passed in Metadata Transfer PDUs are not null terminated.

The Metadata transfer feature adds the following operations that shall be used with PASS THROUGH command:

	Vendor Unique ID	Operation Name	AV/C Command Type	CT	TG	Section
1		Basic Group Navigation				5.6
2.	0x0000	Next Group	CONTROL	M	M	5.6.1
3.	0x0001	Previous Group	CONTROL	M	M	5.6.2

Table 4.9: Operations used with PASS THROUGH command as part of Metadata Transfer Feature

These PASS THROUGH commands shall use BT SIG registered CompanyId as the opcode with the defined vendor unique Id with the PANEL subunit-type. Refer to Appendix-I for packet structure of command and response.

Requirements for CT refer to the ability to send a command.

Requirements for TG refer to the ability to respond to a command.

4.8 Categories

This profile ensures the interoperability by classifying the A/V functions into four categories. For each category, the mandatory commands for the TG are defined by the *operation_id*s in the PASS THROUGH command. It is mandatory for the TG to support at least one of the categories.

4.8.1 Category 1: Player/Recorder

Basic operations of a player or a recorder are defined, regardless of the type of media (tape, disc, solid state, etc.) or the type of contents (audio or video, etc.). If a device supports this category 1, it shall be implemented with the two *operation_id*s of the PASS THROUGH command, “play” and “stop”.

The device supporting category 1 shall support Metadata Information features defined in Table 3.1.

4.8.2 Category 2: Monitor/Amplifier

The category 2 is to define basic operations of a video monitor or an audio amplifier. If a device supports this category 2, it shall be implemented with the two *operation_id*s of the PASS THROUGH command, “volume up” and “volume down”.

4.8.3 Category 3: Tuner

The category 3 defines the basic operation of a video tuner or an audio tuner. If a device supports this category 3, it shall be implemented with the two *operation_id*s of the PASS THROUGH command, “channel up” and “channel down”.

4.8.4 Category 4: Menu

The basic operations for a menu function are defined in category 4. The method to display menu data is not specified. It may be a display panel of the device itself, or on-screen display (OSD) on an external monitor. A device that supports category 4 shall be implemented with the six *operation_id*s of the PASS THROUGH command, “root menu”, “up”, “down”, “left”, “right”, and “select”.

4.8.5 Support Level in TG

The table below is the *operation_id*s and their support level in TG for each category.

“C1” in the table below means that the command is mandatory if the TG supports category 1. In the same manner, “C2” means mandatory in category 2, “C3” in category 3, and “C4” in category 4.

“X” in the table below means that the *operation_id* is not supported in the category.

operation_id	**Category 1:** **Player/Recorder**	**Category 2:** **Monitor/Amplifier**	**Category 3:** **Tuner**	**Category 4:** **Menu**
select	X	X	X	C4
up	X	X	X	C4
down	X	X	X	C4
left	X	X	X	C4
right	X	X	X	C4
right-up	X	X	X	O
right-down	X	X	X	O
left-up	X	X	X	O
left-down	X	X	X	O
root menu	X	X	X	C4
setup menu	X	X	X	O
contents menu	X	X	X	O
favorite menu	X	X	X	O
exit	X	X	X	O
0	O	O	O	O
1	O	O	O	O
2	O	O	O	O
3	O	O	O	O
4	O	O	O	O
5	O	O	O	O
6	O	O	O	O
7	O	O	O	O
8	O	O	O	O
9	O	O	O	O
dot	O	O	O	O
enter	O	O	O	O
clear	O	O	O	O
channel up	X	X	C3	X
channel down	X	X	C3	X

operation_id	Category 1: Player/Recorder	Category 2: Monitor/Amplifier	Category 3: Tuner	Category 4: Menu
previous channel	X	X	O	X
sound select	O	O	O	X
input select	O	O	O	X
display information	O	O	O	O
help	O	O	O	O
page up	X	X	X	O
page down	X	X	X	O
power	O	O	O	O
volume up	X	C2	X	X
volume down	X	C2	X	X
mute	X	O	X	X
play	C1	X	X	X
stop	C1	X	X	X
pause	O	X	X	X
record	O	X	X	X
rewind	O	X	X	X
fast forward	O	X	X	X
eject	O	X	X	X
Forward	O	X	X	X
Backward	O	X	X	X
Angle	O	X	O	X
Subpicture	O	X	O	X
F1	O	O	O	O
F2	O	O	O	O
F3	O	O	O	O
F4	O	O	O	O
F5	O	O	O	O
vendor unique*	O	O	O	O

Table 4.10: Support Levels of operation_id in TG

*: The vendor-unique *operation_id* shall not be used instead of *operation_id* specified in the PASS THROUGH command that has the same functionality.

4.8.6 Support Level in CT

No mandatory command for the CT is defined by the *operation_id*s in the PASS THROUGH command. However, it is mandatory in CT to support at least one of the *operation_id*s. The category for CT indicates that the CT expects to control a TG supporting the corresponding category. It is mandatory for CT to support at least one of categories. The table below is the *operation_id*s and their support level in CT for each category.

“C1” in the table below means that it is mandatory to support at least one of these *operation_id*s if the CT supports category 1. In the same manner, “C2” in category 2, “C3” in category 3, and “C4” in category 4.

“X” in the table below means that the *operation_id* is not supported in the category.

operation_id	**Category 1:** **Player/Recorder**	**Category 2:** **Monitor/Amplifier**	**Category 3:** **Tuner**	**Category 4:** **Menu**
select	X	X	X	C4
up	X	X	X	C4
down	X	X	X	C4
left	X	X	X	C4
right	X	X	X	C4
right-up	X	X	X	C4
right-down	X	X	X	C4
left-up	X	X	X	C4
left-down	X	X	X	C4
root menu	X	X	X	C4
setup menu	X	X	X	C4
contents menu	X	X	X	C4
favorite menu	X	X	X	C4
exit	X	X	X	C4
0	C1	C2	C3	C4
1	C1	C2	C3	C4
2	C1	C2	C3	C4
3	C1	C2	C3	C4
4	C1	C2	C3	C4
5	C1	C2	C3	C4
6	C1	C2	C3	C4
7	C1	C2	C3	C4
8	C1	C2	C3	C4
9	C1	C2	C3	C4
dot	C1	C2	C3	C4
enter	C1	C2	C3	C4
clear	C1	C2	C3	C4
channel up	X	X	C3	X
channel down	X	X	C3	X
previous channel	X	X	C3	X
sound select	C1	C2	C3	X
input select	C1	C2	C3	X
display information	C1	C2	C3	C4
help	C1	C2	C3	C4

operation_id	**Category 1:** **Player/Recorder**	**Category 2:** **Monitor/Amplifier**	**Category 3:** **Tuner**	**Category 4:** **Menu**
page up	X	X	X	C4
page down	X	X	X	C4
power	C1	C2	C3	C4
volume up	X	C2	X	X
volume down	X	C2	X	X
mute	X	C2	X	X
play	C1	X	X	X
stop	C1	X	X	X
pause	C1	X	X	X
record	C1	X	X	X
rewind	C1	X	X	X
fast forward	C1	X	X	X
Eject	C1	X	X	X
Forward	C1	X	X	X
backward	C1	X	X	X
Angle	C1	X	C3	X
subpicture	C1	X	C3	X
F1	C1	C2	C3	C4
F2	C1	C2	C3	C4
F3	C1	C2	C3	C4
F4	C1	C2	C3	C4
F5	C1	C2	C3	C4
vendor unique*	C1	C2	C3	C4

Table 4.11: Support Levels of operation_id in CT

*: The vendor-unique *operation_id* shall not be used instead of *operation_id* specified in the PASS THROUGH command that has the same functionality.

5 Detailed Description

5.1 Capabilities PDUs

5.1.1 GetCapabilities (PDU ID: 0x10)

Description:

This primitive gets the capabilities supported by remote device. This is sent by CT to inquire capabilities of the peer device.

Command format (GetCapabilities with COMPANY_ID as parameter):

Parameters	Size(byte)	Description	Allowed Values
CapabilityID	1 Byte	Specific capability requested.	see Table

Table 5.1: GetCapabilities Command

Allowed values for GetCapabilities Command:

CapabilityID	Value
COMPANY_ID (0x2)	This requests the list of CompanyID supported by TG . All TG devices are expected to send the BT SIG CompanyID as the first supported CompanyID.
EVENTS_SUPPORTED (0x3)	This requests the list of events supported by the TG. TG is expected to respond with all the events supported including the mandatory events defined in this specification.
Other values	Other CapabilityIDs are reserved

Table 5.2: GetCapabilities Command Allowed Values

GetCapabilities Response format for COMPANY_ID:

Parameters	Size(byte)	Description	Allowed Values
CapabilityID	1	Specific capability requested	COMPANY_ID
CapabilityCount (n)	1	Specifies the number of CompanyID returned	1-255
Capability	3…3*n	List of CompanyID	See Table

Table 5.3: GetCapabilities Response for COMPANY_ID

Allowed Values for GetCapabilities Response for COMPANY_ID:

Capability	Value
CompanyID	CompanyID value range as defined in AV/C VENDOR_DEPENDENT command frame. The first COMPANY_ID returned is the BT SIG's defined Metadata Transfer CompanyID. Each CompanyID is 3 bytes long.

Table 5.4: GetCapabilities Response for CompanyID Allowed Values

GetCapabilities Response format for EVENTS_SUPPORTED:

Parameters	Size(byte)	Description	Allowed Values
CapabilityID	1	Specific capability requested	EVENTS_SUPPORTED
CapabilityCount (n)	1	Specifies the number of events supported	2-255
Capability	2…n	List of EventIDs	see Table

Table 5.5: GetCapabilities Response for EVENTS_SUPPORTED

Allowed Values for GetCapabilities Response for EVENTS_SUPPORTED:

Capability	Value
EventIDs	Minimum of two mandatory EventIDs defined in Appendix H: List of defined metadata transfer events is returned. EventIDs are 1 byte each.

Table 5.6: GetCapabilities Response for EVENTS_SUPPORTED Allowed Values

NOTE: The CT should be aware that the capabilities supported by the TG may be subject to change. This may occur if the application on the TG changes, or the application changes mode, for instance different functionality may be available when the TG is playing locally stored audio tracks to when it is acting as a radio. How this is handled by the CT is implementation dependent. If the TG application changes to support less functionality the CT may receive error responses indicating that the function requested is not implemented. The CT may then decide to reissue the GetCapabilities to get the most current capabilities If the TG application changes to support more features the CT may be happy to continue using the original set of features supported. If not it may choose to occasionally poll the TG with a GetCapabilities to determine when further capabilities are available.

5.2 Player application settings PDUs

The following PDUs provide the needed functionality for controller devices to access and set attribute value on the target device.

5.2.1 ListPlayerApplicationSettingAttributes (PDU ID: 0x11)

Description:

This primitive request the target device to provide target supported player application setting attributes. The list of reserved player application setting attributes is provided in Appendix F. It is expected that a target device may have additional attributes not defined as part of this specification.

Command Format (ListPlayerApplicationSettingAttributes)

Parameters	Size(byte)	Description	Allowed Values
None			

Table 5.7: ListPlayerApplicationSettingAttributes command

Response Format (ListPlayerApplicationSettingAttributes)

Parameters	Size(byte)	Description	Allowed Values
NumPlayerApplication SettingAttributes(N)	1	Number of attributes provided	0-255
PlayerApplication SettingAttributeID1	1	Specifies the player application setting attribute ID	See *Appendix F:* List of defined Player Application Settings and Values for the list of player application setting attribute IDs
And so on for the number of target defined player application setting attributes (N).			

Table 5.8: ListPlayerApplicationSettingAttributes response

5.2.2 ListPlayerApplicationSettingValues (PDU ID: 0x12)

Description:

This primitive requests the target device to list the set of possible values for the requested player application setting attribute. The list of reserved player application setting attributes and their values are provided in Appendix F: List of defined Player Application Settings and Values. It is expected that a target device may have additional attribute values not defined as part of this specification.

Command Format (ListPlayerApplicationSettingValues)

Parameters	Size(byte)	Description	Allowed Values
PlayerApplicationSettingAttributeID	1	Player application setting attribute ID	Player application setting attribute ID as defined in *Appendix F:* List of defined Player Application Settings and Values or received from the target

Table 5.9: ListPlayerApplicationSettingValues command

Response Format (ListPlayerApplicationSettingValues)

Parameters	Size(byte)	Description	Allowed Values
NumPlayerApplicationSettingValues (N)	1	Number of player application setting values	1-255
PlayerApplicationSettingValueID1	1	Specifies the player application setting value ID	See *Appendix F:* List of defined Player Application Settings and Values for list of reserved player application setting values. Additional values may be provided by the target for the requested player

			application setting attribute
And so on for the number of target defined player application setting values (N).			

Table 5.10: ListPlayerApplicationSettingValues response

5.2.3 GetCurrentPlayerApplicationSettingValue (PDU ID: 0x13)

Description:

This primitive requests the target device to provide the current set values on the target for the provided player application setting attributes list.

Command Format (GetCurrentPlayerApplicationSettingValue)

Parameters	Size(byte)	Description	Allowed Values
NumPlayerApplicationSettingAttributeID (N)	1	Number of player application setting attribute for which current set values are requested	1-255
PlayerApplicationSettingAttributeID1	1	Player application setting attribute ID for which the corresponding current set value is requested	Valid PlayerApplicationSettingAttributeID values received from the target, or defined as part of *Appendix F:* List of defined Player Application Settings and Values
And so on for the number of target defined player application setting attributes in the requested order (N).			

Table 5.11: GetCurrentPlayerApplicationSettingValue command

Response format (GetCurrentPlayerApplicationSettingValue)

Parameters	Size(byte)	Description	Allowed Values
NumPlayerApplicationSettingValues (N)	1	Number of player application settings value provided	1-255
PlayerApplicationSettingAttributeID1	1	Player application setting attribute ID for which the value is returned	1-255
PlayerApplicationSettingValueID1	1	Currently set player application setting value on the target for the corresponding requested player application setting attribute ID	Valid PlayerApplicationSettingValueID values received from the target, or defined as part of *Appendix F:* List of defined Player Application Settings and Values
And so on for the number of target defined player application setting values in the requested order (N).			

Table 5.12: GetCurrentPlayerApplicationSettingValue response

5.2.4 SetPlayerApplicationSettingValue (PDU ID: 0x14)

Description:

This primitive requests to set the player application setting list of player application setting values on the target device for the corresponding defined list of PlayerApplicationSettingAttributes.

Command Format (SetPlayerApplicationSettingValue)

Parameters	**Size(byte)**	**Description**	**Allowed Values**
NumPlayerApplication SettingAttributes (N)	1	Number of player application setting attributes for which the player application setting	1-255
PlayerApplicationSetting AttributeID1	1	Player application setting attribute ID for which the value needs to be set	Valid PlayerApplicationSettingAttributeID values received from the target, or defined as part of *Appendix F:* List of defined Player Application Settings and Values
PlayerApplication SettingValueID1	1	Player application setting value ID for the corresponding player application setting attribute ID	Valid PlayerApplicationSettingValueID values received from the target, or defined as part of *Appendix F:* List of defined Player Application Settings and Values
And so on for the number of target defined player application setting attributes and their values.			

Table 5.13: SetPlayerApplicationSettingValue command

Response format (SetPlayerApplicationSettingValue)

Parameters	**Size(byte)**	**Description**	**Allowed Values**
None			

Table 5.14: SetPlayerApplicationSettingValue response

NOTE: Setting of a value by CT does not implicitly mean that the setting will take effect on TG. The setting shall take effect after a play command from CT. If currently playing, it is up to the TG to decide when the setting shall take effect. There shall be an error response sent back if there are errors in attribute and/or value. See section 5.7 for additional details.

5.2.5 GetPlayerApplicationSettingAttributeText (PDU ID: 0x15)

Description:

This primitive requests the target device to provide supported player application setting attribute displayable text for the provided PlayerApplicationSettingAttributeIDs.

NOTE: This command is expected to be used only for extended attributes for menu navigation. It is assumed that all <attribute, value> pairs used for menu extensions are statically defined by TG.

Command Format (GetPlayerApplicationSettingAttributeText)

Parameters	**Size(byte)**	**Description**	**Allowed Values**
NumPlayerApplicationSettingAttributes (N)	1	Number of player application setting attribute IDs for which corresponding string is needed	1-255
PlayerApplicationSettingAttributeID1	1	Player application setting attribute ID for which the corresponding attribute displayable text is needed	Valid PlayerApplicationSettingAttributeID values received from the target, or defined attributeID as part of *Appendix F:* List of defined Player Application Settings and Values
And so on for the number of needed player application setting attribute ID (N)			

Table 5.15: GetPlayerApplicationSettingAttributeText command

Response format (GetPlayerApplicationSettingAttributeText)

Parameters	**Size(byte)**	**Description**	**Allowed Values**
NumPlayerApplicationSettingAttributes (N)	1	Number of attributes provided	1-255
PlayerApplicationSettingAttributeID1	1	Specified the player application setting attribute ID for which the displayable text is returned	1-255
CharacterSetID1	2	Specifies the character set ID to be displayed on CT	Use MIBenum defined in IANA character set document (Refer to InformDisplayableCharacterSet)
PlayerApplicationSettingAttributeStringLength1 (n)	1	Length of the player application setting attribute string	1-255
PlayerApplicationSettingAttributeString1	1-n	Specifies the player application setting attribute string in specified character set.	Any string encoded in specified character set
And so on for the number of target defined player application setting attributes in the requested order (N).			

Table 5.16: GetPlayerApplicationSettingAttributeText response

5.2.6 GetPlayerApplicationSettingValueText (PDU ID: 0x16)

Description:

This primitive request the target device to provide target supported player application setting value displayable text for the provided player application setting attribute values.

NOTE: This command is expected to be used only for extended attributes for menu navigation. It is assumed that all <attribute, value> pairs used for menu extensions are statically defined by TG.

Command Format (GetPlayerApplicationSettingValueText)

Parameters	Size(byte)	Description	Allowed Values
PlayerApplicationSettingAttributeID	1	Player application setting attribute ID	Player application setting attribute ID as defined in *Appendix F:* List of defined Player Application Settings and Values or received from the target
NumPlayerApplicationSettingValue(N)	1	Number of player application setting values for which corresponding string is needed	1-255
PlayerApplicationSettingValueID1	1	Player application setting value ID for which the corresponding value string is needed	Valid ValueID values received from the target, or defined ValueID as part of *Appendix F:* List of defined Player Application Settings and Values
And so on for the number of target defined player application setting values in the requested order (N).			

Table 5.17: GetPlayerApplicationSettingValueText command

Response format (GetPlayerApplicationSettingValueText)

Parameters	Size(byte)	Description	Allowed Values
NumPlayerApplicationSettingValues (N)	1	Number of player application settings value provided	1-255
PlayerApplicationSettingValueID1	1	Player application setting value ID for which the text is returned	1-255
CharacterSetID1	2	Specifies the character set ID to be displayed on CT	Refer to section 5.2.7 for allowed values
PlayerApplicationSettingValueStringLength1 (n)	1	Length of the player application setting value string	1-255
PlayerApplicationSettingValueString1	1-n	Specifies the player application setting value string in specified character set.	Any string encoded in specified character set
And so on for the number of target defined player application setting values in the requested order (N).			

Table 5.18: GetPlayerApplicationSettingValueText response

5.2.7 InformDisplayableCharacterSet (PDU ID: 0x17)

Description:

This primitive informs the list of character set supported by CT to TG. This shall allow TG to send responses with strings in the character set supported by CT.

When TG receives this command, the TG can send a string in the character set that is specified in this command. If there is no character set which CT has, TG will send a string in UTF-8. By default TG shall send strings in UTF-8 if this command has not been sent by CT to TG.

Command Format (InformDisplayableCharacterSet)

Parameters	Size(byte)	Description	Allowed Values
NumCharacterSet(N)	1	Number of displayable character sets	1-255
CharacterSetID1	2	Specifies the character set ID to be displayed on CT.	Refer to NOTE for valid values

Table 5.19: InformDisplayableCharacterSet command

Response format (InformDisplayableCharacterSet)

Parameters	Size(byte)	Description	Allowed Values
None			

Table 5.20: InformDisplayableCharacterSet response

Refer to Figure 25.1 in **Appendix J:** List of Example MSC of different Metadata Transfer Commands**.**

Note:

If this command is not issued, UTF-8 shall be used for any strings as default character set. It is mandatory for CT to send UTF-8 as one of the supported character set in the PDU parameters.

The CT should send this command before it sends any commands that support multiple character sets as follows:

- GetPlayerApplicationSettingAttributeText (0x15)
- GetPlayerApplicationSettingValueText (0x16)
- GetElementAttributes (0x20)
- CharacterSetID parameter in all the above listed PDUs including this PDU is MIBenum value of the character set defined in IANA character set document [11].

5.2.8 InformBatteryStatusOfCT (PDU ID: 0x18)

Description:

This command frame is being sent by the CT to TG whenever the CT's battery status has been changed.

Command Format (InformBatteryStatusOfCT)

Parameters	Size(byte)	Description	Allowed Values
Battery status	1	Battery status	0x0 – NORMAL – Battery operation is in normal state 0x1 – WARNING - unable to operate soon. Specified when battery going down. 0x2 – CRITICAL – can not operate any more. Specified when battery going down. 0x3 – EXT ERNAL – Connecting to external power supply 0x4 - FULL_CHARGE – when the device is completely charged.

Table 5.21: InformBatteryStatusOfCT command

Response Format (InformBatteryStatusOfCT)

Parameters	Size(byte)	Description	Allowed Values
None			

Table 5.22: InformBatteryStatusOfCT response

5.3 Media Information PDUs

The Media Information PDU’s are used to obtain detailed information on a particular media file like song information including title, album, artist, composer, year etc.

5.3.1 GetElementAttributes (PDU ID: 0x20)

Description:

These primitive requests the TG to provide the attributes of the element specified in the parameter.

Command Format (GetElementAttributes)

Parameters	Size(byte)	Description	Allowed Values
Identifier	8	Unique identifier to identify an element on TG	PLAYING (0x0): This should return attribute information for the element which is current track in the TG device. All other values other than 0x0 are currently reserved.
NumAttributes (N)	1	Number of Attributes provided	If NumAttributes is set to zero, all attribute information shall be returned, else attribute information for the specified attribute IDs shall be returned by the TG
AttributeID1	4	Specifies the attribute ID for the attributes to be retrieved	See Appendix E: List of Media Attributes for the list of possible attribute IDs
And so on for each attribute (1…N).			

Table 5.23: GetElementAttributes command

Response Format (GetElementAttributes):

Parameters	Size(byte)	Description	Allowed Values
NumAttributes (N)	1	Number of attributes provided	1-255
AttributeID1	4	Specifies the attribute ID to be written	See Appendix E: List of Media Attributes for list of possible attribute IDs
CharacterSetID1	2	Specifies the character set ID to be displayed on CT	Use MIBenum defined in IANA character set document [11] (Refer to InformDisplayableCharacterSet)
AttributeValueLength1 (n1)	2	Length of the value of the attribute	0-65535 (0, if no name is provided)
AttributeValue1	1-n1	Attribute Name in specified character set	Any text encoded in specified character set
And so on for all the attributes provided (1…N)			

Table 5.24: GetElementAttributes response

5.4 Notification PDUs

The Notification PDUs are used to obtain synchronous as well as asynchronous updates from the TG based on change of status at the target's side.

For example, when CT might be interested to know the current status of a media track or when media track gets changed, so that new media information can be displayed on the controller's display. The CT could do one of i) querying for play status or ii) register with the TG to receive play status notifications. The TG then sends a notification PDU when a status change happens if the CT had registered for that change.

5.4.1 GetPlayStatus (PDU ID: 0x30)

Description:

This primitive is used by the CT to get the status of the currently playing media at the TG.

Command Format (GetPlayStatus) :

Parameters	Size(byte)	Description	Allowed Values
None			

Table 5.25: GetPlayStatus command

Response Format (GetPlayStatus):

Parameters	Size(byte)	Description	Allowed Values
SongLength	4	The total length of the playing song in milliseconds	0-(2^{32} – 1)
SongPosition	4	The current position of the playing in milliseconds elapsed	0-(2^{32} – 1)
PlayStatus	1	Current Status of playing	0x00 : STOPPED 0x01 : PLAYING 0x02 : PAUSED 0x03: FWD_SEEK 0x04: REV_SEEK 0xFF : ERROR

Table 5.26: GetPlayStatus response

Note:

If TG does not support SongLength And SongPosition on TG, then TG shall return 0xFFFFFFFF.

5.4.2 RegisterNotification (PDU ID: 0x31)

Description:

This primitive registers with the TG to receive notifications asynchronously based on specific events occurring. The initial response to this Notify command shall be an INTERIM response with current status, or a REJECTED/NOT IMPLEMENTED response. This has to take place within T_{MTP} time from receiving the command. The following response shall be a CHANGED response with the updated status, or a

REJECT response. This is as per 1394 AV/C protocol specification. A registered notification gets changed on receiving CHANGED event notification. For a new notification additional NOTIFY command is expected to be sent. Only one EventID shall be used per notification registration.

Refer to Figure 25.2 in Appendix J: List of Example MSC of different Metadata Transfer Commands.

Command Format (RegisterNotification):

Parameters	**Size(byte)**	**Description**	**Allowed Values**
EventID	1	Event for which the CT requires notifications	see **Error! Reference source not found.**
Playback interval	4	Specifies the time interval (in seconds) at which the change in playback position will be notified. If the song is being forwarded / rewound, a notification will be received whenever the playback position will change by this value. (Applicable only for EventID EVENT_PLAYBACK_POS_CHANGED. For other events , value of this parameter is ignored)	0 < Playback interval

Table 5.27: RegisterNotification command

Allowed Values for EventID

EventID	Description
EVENT_PLAYBACK_STATUS_CHANGED (0x01)	Event for change in playback status
EVENT_TRACK_CHANGED (0x02)	Event for change in track
EVENT_TRACK_REACHED_END (0x03)	Event for reach to end of the current track
EVENT_TRACK_REACHED_START (0x04)	Event for reach to start of the current track.
EVENT_PLAYBACK_POS_CHANGED (0x05)	Event for change in playback position
EVENT_BATT_STATUS_CHANGED (0x06)	Event for change in battery status
EVENT_SYSTEM_STATUS_CHANGED (0x07)	Event for change in system status
EVENT_PLAYER_APPLICATION_SETTING_CHANGED (0x08)	Event for change in player application setting

Table 5.28: Allowed Values for EventID

Response Formats (RegisterNotification)

Response Data format for EVENT_PLAYBACK_STATUS_CHANGED

Parameters	Size(byte)	Description	Allowed Values
EventID	1	Specific EventID	EVENT_PLAYBACK_STATUS_CHANGED (0x01)
PlayStatus	1	Indicates the current status of playback	0x00: STOPPED 0x01: PLAYING 0x02: PAUSED 0x03: FWD_SEEK 0x04: REV_SEEK 0xFF: ERROR

Table 5.29: Response EVENT_PLAYBACK_STATUS_CHANGED

Response Data format for EVENT_TRACK_CHANGED

Parameters	Size(byte)	Description	Allowed Values
EventID	1	Specific EventID	EVENT_TRACK_CHANGED(0x02)
Identifier	8	Index of the current track	If no track currently selected, then return 0xFFFFFFFF in the INTERIM response.

Table 5.30: Response EVENT_TRACK_CHANGED

Response Data format for EVENT_TRACK_REACHED_END

Parameters	Size(byte)	Description	Allowed Values
EventID	1	Specific EventID	EVENT_TRACK_REACHED_END(0x03)
None			

Table 5.31: Response EVENT_TRACK_REACHED_END

Response Data format for EVENT_TRACK_REACHED_START

Parameters	Size(byte)	Description	Allowed Values
EventID	1	Specific EventID	EVENT_TRACK_REACHED_START (0x04)
None			

Table 5.32: Response EVENT_TRACK_REACHED_START

Response Data format for EVENT_ PLAYBACK_POS_CHANGED

Parameters	Size(byte)	Description	Allowed Values
EventID	1	Specific EventID	EVENT_PLAYBACK_POS_CHANGED (0x05)
Playback position	4	Current playback position in millisecond	If no track currently selected, then return 0xFFFFFFFF in the INTERIM response.

Table 5.33: Response EVENT_ PLAYBACK_POS_CHANGED

EVENT_PLAYBACK_POS_CHANGED shall be notified in the following conditions:

- TG has reached the registered playback Interval time.
- Changed PLAY STATUS.
- Changed Current Track.
- Reached end or beginning of track.

Response Data format for EVENT_BATT_STATUS_CHANGED

Parameters	Size(byte)	Description	Allowed Values
EventID	1	Specific EventID	EVENT_BATT_STATUS_CHANGED (0x06)
Battery status	1	Battery status	see **Error! Reference source not found.**

Table 5.34: Response EVENT_BATT_STATUS_CHANGED

Allowed Values for Battery Status:

Battery Status Value	Description
0x0 – NORMAL –	Battery operation is in normal state
0x1 – WARNING -	unable to operate soon. Is provided when the battery level is goind down.

Battery Status Value	Description
0x2 – CRITICAL –	can not operate any more. Is provided when the battery level is going down.
0x3 – EXTERNAL –	Plugged to external power supply
0x4 - FULL_CHARGE –	when the device is completely charged from the external power supply

Table 5.35: Allowed Values for Battery Status

NOTE: Battery status notification defined in this specification is expected to be replaced by Attribute profile specification in the future.

Response Data format for EVENT_SYSTEM_STATUS_CHANGED

Parameters	Size(byte)	Description	Allowed Values
EventID	1	Specific EventID	EVENT_SYSTEM_STATUS_CHANGED (0x07)
SystemStatus	1	Indicates the current System status.	POWER_ON (0x00) POWER_OFF (0x01) UNPLUGGED (0x02)

Table 5.36: Response EVENT_SYSTEM_STATUS_CHANGED

POWER_OFF and UNPLUGGED are used for Bluetooth Accessories which attach to Media Players. In this case, it will happen that Audio Player's power state is "POWER OFF" or Audio Player is detached from Bluetooth Adapter (UNPLUGGED)

Response Data format for EVENT_ PLAYER_APPLICATION_SETTING_CHANGED

Parameters	Size(byte)	Description	Allowed Values
EventID	1	Specific EventID	EVENT_ PLAYER_APPLICATION_SETTING_CHANGED (0x08)
NumPlayerApplication SettingAttributes(N)	1	Number of player application setting attributes that follow	1-255
PlayerApplicationSettingAttributeID1	1	Player application setting attribute ID for which the value is returned	1-255
PlayerApplicationSettingValueID1	1	Currently set player application setting value on the target for the corresponding requested PlayerApplicationSettingAttributeID	Valid PlayerApplicationSettingValueID values, or defined as part of Appendix F
And so on for the number of target defined player application setting values.			

Table 5.37: Response EVENT_ PLAYER_APPLICATION_SETTING_CHANGED

5.5 Continuation PDUs

5.5.1 RequestContinuingResponse (PDU ID: 0x40)

Description:

This primitive is used by CT to request for continuing response packets for the sent PDU command, that has not completed. This command will be invoked by CT after receiving a response with <Packet Type – First (0x01) or Continue (0x10)>.

Command Format (RequestContinuingResponse)

Parameters	Size(byte)	Description	Allowed Values
ContinuePDU_ID	1	Target PDU_ID for continue command	PDU_ID

Table 5.38: RequestContinuingResponse command

Response Format (RequestContinuingResponse)

The response for this command is the pending data for the previous command invoked by CT. Refer to Figure 25.3 in Appendix J: List of Example MSC of different Metadata Transfer Commands. See also section 4.7.2.

5.5.2 AbortContinuingResponse (PDU ID: 0x41)

Description:

This primitive is used by CT to abort continuing response. This command will be invoked by CT after receiving a response with <Packet Type – First (0x01) or Continue (0x10)>. Refer to Figure 25.4 in Appendix J.

Command Format (AbortContinuingResponse)

Parameters	Size(byte)	Description	Allowed Values
ContinueAbort PDU_ID	1	Target PDU_ID for abort continue command	PDU_ID

Table 5.39: AbortContinuingResponse command

Response Format (AbortContinuingResponse)

Parameters	Size(byte)	Description	Allowed Values
None			

Table 5.40: AbortContinuingResponse response

5.6 Basic Group Navigation

Basic group navigation PDUs are defined to support a logical one dimensional group structure of media content on the TG to CT for easier navigation purpose. The definition of groups on the TG is implementation dependent. The group structure can consist of parts of, or a mix of playlists and artist/album/genre folders etc that are used by the media player applications in the TG.

The basic group navigation PDUs have a similar behavior as the Forward and Backward commands, but instead of navigating to the next/previous song they are used to navigate to the first song in the next/previous group.

The Basic Group Navigation PDUs are transported as vendor unique PASS THROUGH commands. Refer to Appendix-I for vendor unique PDU format used for Basic Group Navigation.

5.6.1 Next Group (vendor unique id: 0x00)

Description:

This function is used to move to the first song in the next group.

5.6.2 Previous Group (vendor unique id: 0x01)

Description:

This function is used to move to the first song in the previous group.

5.7 Error handling for Metadata Transfer Commands

If CT sent a PDU with nonexistent PDU ID or a PDU containing only one parameter with nonexistent parameter ID, TG shall return REJECTED response with Error Status Code. TG may return REJECTED response also in other situations (See 5.7.1 Error Status Code).

If CT sent a PDU with multiple parameters where at least one ID is existent and the others are nonexistent, TG shall proceed with the existent ID and ignore the non-existent IDs.

Note, that CT can always have complete information which IDs were accepted by TG: in case of STATUS PDUs the response will contain information for the IDs which were understood, when setting values for Player application settings; TG will return notification response with the list of AttributeIDs for which values have been set.

5.7.1 Error Status Code

An error status code is added to the REJECTED response if TG rejected the command. It is useful for CT to know why the command is rejected by TG. Table 5.41 shows the error status code.

Error Code Value	Description
0x00	Invalid command, sent if TG received a PDU that it did not understand.
0x01	Invalid parameter, sent if the TG received a

	PDU with a parameter ID that it did not understand. Sent if there is only one parameter ID in the PDU.
0x02	Specified parameter not found., sent if the parameter ID is understood, but content is wrong or corrupted.
0x03	Internal Error, sent if there are other error conditions.

Table 5.41: List of Error Status Code

An example of response packet format for REJECTED will be as below.

Oct	MSB (7)	6	5	4	3	2	1	LSB (0)
0	0x0				Response: 0xA(REJECTED)			
1	Subunit_type:0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x0 (VENDOR DEPENDENT)							
3 – 5	Company ID: [BT SIG specified CompanyID]							
6	PDU ID (of the command for which this response is sent)							
7	Reserved (0x00)						Packet Type (0x0)	
8 – 9	Parameter Length (0x0001)							
10	Error Code (0x02 – Specified parameter not found)							

Table 5.42: An example of response packet format for REJECTED

6 Service Discovery Interoperability Requirements

This profile defines the following service records for the CT and the TG, respectively.

The codes assigned to the mnemonics used in the Value column as well as the codes assigned to the attribute identifiers (if not specifically mentioned in the AttrID column) can be found in the Bluetooth Assigned Numbers document [6].

Item	Definition	Type	Value	AttrID	Status	Default
Service Class ID List					M	
Service Class #0		UUID	A/V Remote Control		M	
Protocol Descriptor List					M	
Protocol #0		UUID	L2CAP		M	
Parameter #0 for Protocol #0	PSM	Uint 16	PSM= AVCTP		M	
Protocol #1		UUID	AVCTP		M	
Parameter #0 for Protocol #1	Version	Uint 16	0x0102*[1]		M	
Bluetooth Profile Descriptor List					M	
Profile #0		UUID	A/V Remote Control		M	
Parameter #0 for Profile #0	Version	Uint 16	0x0103*[2]		M	
Supported Features	AVRCP features flags	Uint 16	*3) Bit 0 = Category 1 Bit 1 = Category 2 Bit 2 = Category 3 Bit 3 = Category 4 Bit 4-15 = RFA The bits for supported categories are set to 1. Others are set to 0.		M	
Provider Name	Displayable Text Name	String	Provider Name		O	
Service Name	Displayable Text Name	String	Service Provider-defined		O	

Table 6.1: Service Record for CT

*1: The value indicates Version 1.2.
*2: The value indicated Version 1.3.

*3: The value indicates the category(ies) of a TG that the CT expects to control. It is not necessary for a CT to have capabilities to initiate all of the mandatory commands of the indicated category(ies).

Item	Definition	Type	Value	AttrID	Status	Default
Service Class ID List					M	
Service Class #0		UUID	A/V Remote Control Target		M	
Protocol Descriptor List					M	
Protocol #0		UUID	L2CAP		M	
Parameter #0 for Protocol #0	PSM	Uint 16	PSM=AVCTP		M	
Protocol #1		UUID	AVCTP		M	
Parameter #0 for Protocol #1	Version	Uint 16	0x0102*[1]		M	
Bluetooth Profile Descriptor List					M	
Profile #0		UUID	A/V Remote Control		M	
Parameter #0 for Profile #0	Version	Uint 16	0x0103*[1]		M	
Supported Features	AVRCP features flags	Uint 16	*2 Bit 0 = Category 1 Bit 1 = Category 2 Bit 2 = Category 3 Bit 3 = Category 4 Bit 4 = Player Application Settings. Bit 0 should be set for this bit to be set. Bit 5 = Group Navigation. Bit 0 should be set for this bit to be set. Bit 6-15 = RFA The bits for supported categories are set to 1. Others are set to 0.		M	
Provider Name	Displayable Text Name	String	Provider Name		O	

Item	Definition	Type	Value	AttrID	Status	Default
Service Name	Displayable Text Name	String	Service-provider defined		O	

Table 6.2: Service Record for TG

*1: The value indicates Version 1.3.

*2: The value indicates the category(ies) that the TG supports. The TG shall be implemented with all of mandatory commands of the indicated category(ies).

7 L2CAP Interoperability Requirements

The following text together with the associated sub-clauses defines the mandatory requirements with regard to this profile.

	Procedure	Support in CT/TG
1.	Channel types	
	Connection-oriented channel	M
	Connectionless channel	X1
2.	Signaling	
	Connection establishment	M
	Configuration	M
	Connection Termination	M
	Echo	M
	Command Rejection	M
3.	Configuration Parameter Options	
	Maximum Transmission Unit	M
	Flush Timeout	M
	Quality of Service	O

X1: Connectionless channel is not used within the execution of this profile, but concurrent use by other profiles/applications is not excluded.

7.1 Channel Types

In this profile, only connection-oriented channels shall be used. This implies that broadcasts shall not be used in this profile.

In the PSM field of the Connection Request packet, the value for AVCTP defined in the Bluetooth Assigned Numbers document [6] shall be used.

7.2 Signaling

AVRCP does not impose any restrictions or requirements on L2CAP signaling.

7.3 Configuration Options

This section describes the usage of configuration options in AVRCP.

7.3.1 Maximum Transmission Unit

The minimum MTU that a L2CAP implementation for this profile shall support is 48 bytes.

7.3.2 Flush Timeout

Application shall set the appropriate value for responding time to the flush timeout.

Remark: Flush timeout can be constrained by the ACL channels when the other applications (such as audio/video streaming or file sharing) coexist with AVRCP.

7.3.3 Quality of Service

Negotiation of Quality of Service is optional in this profile.

8 Link Manager (LM) Interoperability Requirements

The procedure for SCO links is excluded. Other than that, there is no change to the requirements as stated in the Link Manager specification itself. (See Section 3 in [9].)

9 Link Controller (LC) Interoperability Requirements

The following table lists all features at LC level, and the extra requirements are added to the one in the Baseband specification by this profile.

	Procedure	Support in CT	Support in TG
1.	Inquiry	M	O
2.	Inquiry scan	M	M
3.	Paging	M	O
4.	Page scan		
	A. Type R0	C1	C2
	B. Type R1	C1	C2
	C. Type R2	C1	C2
5.	Packet types		
	A. ID packet	M	M
	B. NULL packet	M	M
	C. POLL packet	M	M
	D. FHS packet	M	M
	E. DM1packet	M	M
	F. DH1 packet	M	M
	G. DM3 packet	O	O
	H. DH3 packet	O	O
	I. DM5 packet	O	O
	J. DH5 packet	O	O
	K. AUX packet	X	X
	L. HV1 packet	X	X
	M. HV2 packet	X	X
	N. HV3 packet	X	X
	O. DV packet	X	X
6.	Inter-piconet capabilities	X	X
7.	Air mode		
	A. A-law	X	X
	B. μ-law	X	X
	C. CVSD	X	X
	D. Transparent data	X	X

Table 9.1: LC Capabilities

C1, C2: It is mandatory to implement at least one of the page scan modes.

9.1 Class of Device

A device that is active in the CT role shall indicate as follows in the Class of Device field, if it is a stand-alone remote controller.

1. Indicate ‘Peripheral’ as Major Device class
2. Indicate “Remote control” as the Minor Device class

10 Generic Access Profile Requirements

This section defines the support requirements for the capabilities as defined in the Generic Access Profile [8].

10.1 Modes

The table shows the support status for Modes within this profile.

	Procedure	Support in CT	Support in TG
1.	Discoverability modes		
	Non-discoverable mode	C1	C1
	Limited discoverable mode	O	O
	General discoverable mode	M	M
2.	Connectability modes		
	Non-connectable mode	X	X
	Connectable mode	M	M
3.	Pairing modes		
	Non-pairable mode	O	O
	Pairable mode	C2	C2

Table 10.1: Modes

C1: If Limited discoverable mode is supported, Non-discoverable mode is mandatory otherwise optional.

C2: Mandatory if Bonding is supported otherwise optional.

10.2 Security Aspects

There is no change to the requirements as stated in the General Access Profile.

10.3 Idle Mode Procedures

The table shows the support status for Idle mode procedures within this profile.

	Procedure	**Support in CT**	**Support in TG**
1.	General inquiry	M	O
2.	Limited inquiry	O	O
3	Name discovery	O	O
4.	Device discovery	O	O
5.	Bonding	O	O*

Table 10.2: Supported Idle Mode Procedures

*: Acceptance of bonding shall be supported. If General inquiry is supported, initiation of bonding shall be supported, otherwise, should be supported.

11 Timers and Counters

The following timer is required by AVRCP.

Timer Name	Proposed Value	Description
T_{RCP} (100)	100 milliseconds	A TG shall return its response frame within 100 milliseconds counting from the receipt of the command frame.
T_{MTC} (200)	200 milliseconds	A TG shall return its response frame within 200 milliseconds counting from the receipt of the command frame.
T_{MTP} (1000)	1000 milliseconds	A TG shall return its response frame within 1000 milliseconds counting from the receipt of the command frame.

Table 11.1: Timers

There are no AVRCP specific counters.

12 Testing

The Audio Video Remote Control Profile requires interoperability test. The details of the test strategy are described in [5]. Tested functionality is defined in [4].

13 References

[1] 1394 Trade Association , AV/C Digital Interface Command Set – General Specification, Version 4.0, Document No. 1999026 and AV/C Digital Interface Command Set - General Specification, Version 4.1, Document No. 2001012 (http://www.1394ta.org)

[2] 1394 Trade Association , AV/C Panel Subunit, Version 1.1, Document No. 2001001 (http://www.1394ta.org)

[3] Bluetooth SIG, Specification of the Bluetooth System, Profiles, Version 1.0 or Later, Audio/Video Control Transport Protocol

[4] Bluetooth SIG, Specification of the Bluetooth System, ICS, Version 1.0 or Later, ICS proforma for Audio/Video Remote Control Profile

[5] Bluetooth SIG, Specification of the Bluetooth System, TSS, Version 1.0 or Later, Test Suite Structure (TSS) and Test Procedures (TP) for Audio/Video Remote Control Profile

[6] Bluetooth SIG, Bluetooth Assigned Numbers http://www.bluetooth.org/

[7] Bluetooth SIG, Specification of the Bluetooth System, Core, Version 1.2 or Later, Baseband

[8] Bluetooth SIG, Specification of the Bluetooth System, Core, Version 1.2 or Later, Generic Access Profile

[9] Bluetooth SIG, Specification of the Bluetooth System, Core, Version 1.2 or Later, Link Manager Protocol

[10] Bluetooth SIG, Specification of the Bluetooth System, Core, Version 1.2 or Later, Service Discovery Protocol

[11] http://www.iana.org/assignments/character-sets

14 List of Figures

15 List of Tables

16 Appendix A (Informative): Example of Latency

This section is intended to be information for application only: There are no requirements for the latency.

The value of maximum latency is shown below.

The latency includes the initiation on the sender side up to the start of the requested procedure on the receiving side.

Application example	From	To	Latency
Figure 2.3: Remote Control from Separate Controller	Remote Controller	Portable Disc Player	200 msec
Figure 2.4: Remote Control and Audio Stream between Two Devices	Headphone	Portable Disc Player	200 msec
Figure 2.5: Mutual Remote Control within a Piconet	Headphone	Portable Disc Player	200 msec
Figure 2.6: Headphone with LCD connected to media player	Headphone with LCD Remote Controller	Portable Disc Player	100 msec
Figure 2.6: Headphone with LCD connected to media player	Remote Controller	VCR	100 msec

Table 16.1 Example of Latency

17 Appendix B (Informative): Example of A/V Devices

General functions of A/V devices can be realized by choosing several categories from category 1 to category 4 of TG. The following table shows the possible combination of categories for each function. Note that the table simply presents examples, and does not specify categories that a device shall support.

Functions	**Categories to Support**	**Device Examples**
Audio player without volume control	1	CD player (component), MD player (component)
Audio player with volume control	1, 2	portable disk player
Audio receiver	3	tuner (component)
Audio receiver	2, 3	portable radio
Audio recorder with receiver	1, 2, 3	cassette tape recorder with receiver
Audio amplifier	2	amplifier, headphone
Video recorder without volume control	1	portable video camera recorder
Video recorder with volume control	1, 2, 3	portable VCR with LCD display, TV with VCR
Video recorder with receiver	1, 3	VCR, video disk recorder
TV	2, 3	TV
Video recorder with menu operation	1, 3, 4	VCR with menu control function
TV with menu operation	2, 3, 4	TV with menu control function
Amplifier with menu operation	2, 4	amplifier with menu control function
Video monitor with menu operation	4	video projector with menu control function

Table 17.1: Category Combination Examples

18 Appendix C (Informative): Multiple applications use of AVCTP

Every profile based on Audio/Video Control Transport Protocol (AVCTP) uses a single L2CAP channel. When there are two devices, one simply works as the CT and another simply as the TG; the connection on a single L2CAP channel between them can be established or released by an application as the need arises. However, when one of the devices supports several profiles or two roles, the CT and the TG, the operation to release a connection should be manipulated carefully.

For example, even if application ‘A’ wants to discard a connection for control, another application ‘B’ may need the connection kept established. If application ‘A’ releases the connection on its own judgment, and then if application ‘B’ needs to send a command, application ‘B’ shall re-establish another connection for control to send a command, which causes a delay.

A necessary connection to be released by another application can be avoided by implementation. That is, before releasing the connection for control, an application should try to investigate whether other profiles or other role of the same profile in the device uses AVCTP. It is recommended to apply above implementation solution when developing a device that supports both CT and TG, or supports another control profile in addition to AVRCP.

19 Appendix D (Informative): Example of AV/C Commands and Responses

This chapter shows several examples of commands from a CT and responses from a TG exchanged in case a TG supports only AVRCP as its AV control profile. Note that the structures of commands and responses mentioned in this chapter are merely examples, and fields may have different structures or values according to the situations. Refer AV/C General Specification [1] and AV/C Panel Subunit Specification [2].

19.1 UNIT INFO command

The frame structure of UNIT INFO command is as shown below.

msb	lsb	
0 0 0 0	ctype: STATUS (1_{16})	octet 0
subunit_type: Unit ($1F_{16}$)	subunit_ID: Ignore(7_{16})	octet 1
opcode: UNIT INFO (30_{16})		octet 2
(FF_{16})		octet 3
(FF_{16})		octet 4
(FF_{16})		octet 5
(FF_{16})		octet 6
(FF_{16})		octet 7

Figure 19.1: UNIT INFO Command Frame

An example of a response returned to above command frame is as follows.

msb	lsb	
0 0 0 0	response: STABLE (C_{16})	octet 0
subunit_type: Unit ($1F_{16}$)	subunit_ID: Ignore(7_{16})	octet 1
opcode: UNIT INFO (30_{16})		octet 2
(07_{16})		octet 3
unit_type: Panel (9_{16})	Unit: (0_{16})	octet 4
company_ID ($XXXXXX_{16}$)		octet 5 octet 6 octet 7

Figure 19.2: UNIT INFO Response Frame

If, in future, a Bluetooth AV control profile that applies AV/C command set is defined, and if a TG supports this AV control profile in addition to AVRCP, it is possible that a TG returns other subunit type than Panel as its *unit_type*.

19.2 SUBUNIT INFO command

The frame structure of SUBUNIT INFO command is as shown below.

msb			lsb	
0 0 0 0		ctype: STATUS (1_{16})		octet 0
subunit_type: Unit ($1F_{16}$)		subunit_ID: Ignore(7_{16})		octet 1
opcode: SUBUNIT INFO (31_{16})				octet 2
0	page: (0_{16})	0	extention_code: (7_{16})	octet 3
(FF_{16})				octet 4
(FF_{16})				octet 5
(FF_{16})				octet 6
(FF_{16})				octet 7

Figure 19.3: SUBUNIT INFO Command Frame

An example of a response returned to above command frame is as follows.

msb			lsb	
0 0 0 0		response: STABLE (C_{16})		octet 0
subunit_type: Unit ($1F_{16}$)		subunit_ID: Ignore(7_{16})		octet 1
opcode: SUBUNIT INFO (31_{16})				octet 2
0	page: (0_{16})	0	extention_code: (7_{16})	octet 3
subunit_type: Panel (9_{16})		max_subunit_ID: (0_{16})		octet 4
(FF_{16})				octet 5
(FF_{16})				octet 6
(FF_{16})				octet 7

Figure 19.4: SUBUNIT INFO Response Frame

If, in future, a Bluetooth AV control profile that applies AV/C command set is defined, and if a TG supports this AV control profile in addition to AVRCP, the TG returns all of its supporting subunits including Panel in *page_data* field.

19.3 PASS THROUGH command

The PASS THROUGH command is a command sent when a “PLAY” button on a CT is pushed by a user. Its frame structure is as shown below. A CT sends a command frame with its *state_flag* field in value 0 when a button is pushed, and in value 1 when the button is released.

msb		lsb
0 0 0 0	ctype: CONTROL (0_{16})	octet 0
subunit_type: Panel (9_{16})	subunit_ID: (0_{16})	octet 1
opcode: PASS THROUGH ($7C_{16}$)		octet 2
0	operation_ID: play (44_{16})	octet 3
operation_data_field_length: (0_{16})		octet 4

Figure 19.5: PASS THROUGH Command Frame

An example of a response returned to above command frame is as follows.

msb		lsb
0 0 0 0	response: ACCEPTED (9_{16})	octet 0
subunit_type: Panel (9_{16})	subunit_ID: (0_{16})	octet 1
opcode: PASS THROUGH ($7C_{16}$)		octet 2
0	operation_ID: play (44_{16})	octet 3
operation_data_field_length: (0_{16})		octet 4

Figure 19.6: PASS THROUGH Response Frame

20 Appendix E: List of Media Attributes

The table below provides the list of IDs for Attributes. These IDs are used to uniquely identify media information. Additional information on Media Attributes is available in Bluetooth Assigned Numbers [6].

Attribute ID	Description	Allowed values	Mandatory/ Optional
0x0	Illegal , Should not be used	-	
0x1	Title of the media	Any text encoded in specified character set	M
0x2	Name of the artist	Any text encoded in specified character set	O
0x3	Name of the album	Any text encoded in specified character set	O
0x4	Number of the media (ex. Track number of the CD)	Numeric ASCII text with zero suppresses	O
0x5	Total number of the media (ex. Total track number of the CD)	Numeric ASCII text with zero suppresses	O
0x6	Genre	Any text encoded in specified character set	O
0x7	Playing time in millisecond	Numeric ASCII text with zero suppresses (Ex. 2min30sec → 150000)	O
0x8-0xFFFFFFFF	Reserved for future use	-	

Table 20.1: Attribute IDs

NOTE: If the track title is not available the TG shall try to identify the track in other ways or send information about the media. If no information is available an empty string of zero length may be sent.

21 Appendix F: List of defined Player Application Settings and Values

The table below provides the list of IDs for player application settings. These IDs are used to uniquely identify and exchange information on player application settings between the TG and the CT. Additional information on Player Application Settings is available in Bluetooth Assigned Numbers [6].

<table>
<tr><th>Player Application Setting Attribute</th><th>Attribute Description</th><th>Defined Values</th><th>M/O</th></tr>
<tr><td>0x00</td><td>Illegal , Should not be used</td><td>None</td><td>O</td></tr>
<tr><td>0x01</td><td>Equalizer ON/OFF status</td><td>PlayerApplicationSettingValueID
<table>
<tr><th>ValueID</th><th>Description</th></tr>
<tr><td>0x01</td><td>OFF</td></tr>
<tr><td>0x02</td><td>ON</td></tr>
<tr><td>0x03-0xFF</td><td>Reserved for future use</td></tr>
</table></td><td>O</td></tr>
<tr><td>0x02</td><td>Repeat Mode status</td><td>PlayerApplicationSettingValueID
<table>
<tr><th>ValueID</th><th>Description</th></tr>
<tr><td>0x01</td><td>OFF</td></tr>
<tr><td>0x02</td><td>Single track repeat</td></tr>
<tr><td>0x03</td><td>All track repeat</td></tr>
<tr><td>0x04</td><td>Group repeat</td></tr>
<tr><td>0x05-0xFF</td><td>Reserved for future use</td></tr>
</table></td><td>O</td></tr>
<tr><td>0x03</td><td>Shuffle ON/OFF status</td><td>PlayerApplicationSettingValueID
<table>
<tr><th>ValueID</th><th>Description</th></tr>
<tr><td>0x01</td><td>OFF</td></tr>
<tr><td>0x02</td><td>All tracks shuffle</td></tr>
<tr><td>0x03</td><td>Group shuffle</td></tr>
<tr><td>0x04-0xFF</td><td>Reserved for future use</td></tr>
</table></td><td>O</td></tr>
<tr><td>0x04</td><td>Scan ON/OFF status</td><td>PlayerApplicationSettingValueID
<table>
<tr><th>ValueID</th><th>Description</th></tr>
<tr><td>0x01</td><td>OFF</td></tr>
<tr><td>0x02</td><td>All tracks scan</td></tr>
</table></td><td>O</td></tr>
</table>

<table>
<tr><td></td><td></td><td><table><tr><td>0x03</td><td>Group scan</td></tr><tr><td>0x04-0xFF</td><td>Reserved for future use</td></tr></table></td><td></td></tr>
<tr><td>0x05 – 0x7F</td><td>Reserved for future use</td><td></td><td>O</td></tr>
<tr><td>0x80 – 0xFF</td><td>Provided for TG driven static media player menu extension by CT</td><td></td><td>O</td></tr>
</table>

Table 21.1: PlayerApplicationSettingAttributeIDs

22 Appendix G (Informative): Example MSC for extracting metadata transfer information from TG

Message Sequence Chart (MSC)

Below is an example MSC on how to access track information from TG when a track change event occurrs due to a PASSTHROUGH command from CT.

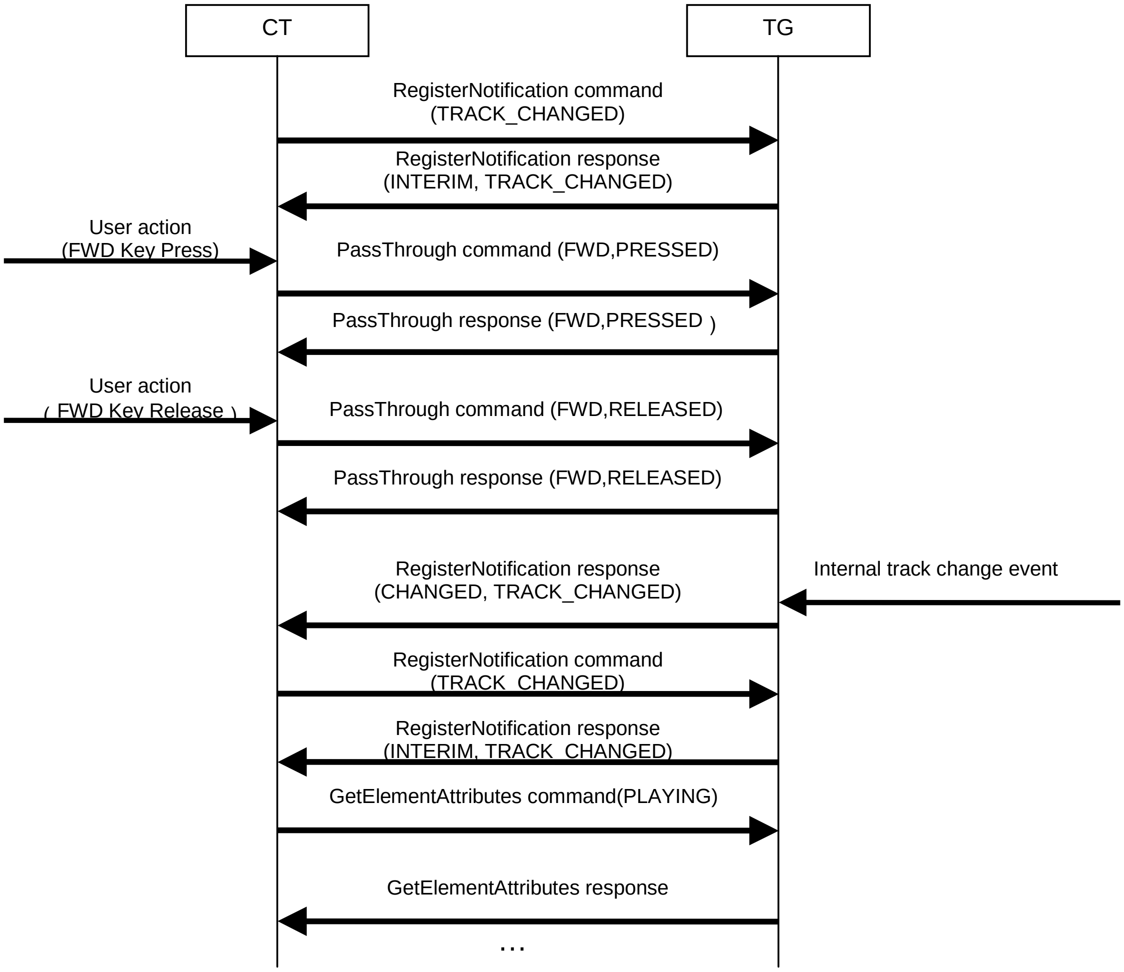

CT
TG
RegisterNotification command (TRACK_CHANGED)
RegisterNotification response (INTERIM, TRACK_CHANGED)
User action (FWD Key Press)
PassThrough command (FWD,PRESSED)
PassThrough response (FWD,PRESSED)
User action (FWD Key Release)
PassThrough command (FWD,RELEASED)
PassThrough response (FWD,RELEASED)
RegisterNotification response (CHANGED, TRACK_CHANGED)
Internal track change event
RegisterNotification command (TRACK_CHANGED)
RegisterNotification response (INTERIM, TRACK_CHANGED)
GetElementAttributes command(PLAYING)
GetElementAttributes response
…

Figure 22.1 Example Message Sequence Chart

23 Appendix H: List of defined metadata transfer events

The table below gives the list of EventIDs defined in this specification to be supported by TG. Additional information on EventIDs is available in Bluetooth Assigned Numbers [6].

EventID	**Description**	**Mandatory /Optional**
EVENT_PLAYBACK_STATUS_CHANGED (0x01)	Change in playback status of the current track.	M
EVENT_TRACK_CHANGED (0x02)	Change of current track	M
EVENT_TRACK_REACHED_END (0x03)	Reached end of a track	O
EVENT_TRACK_REACHED_START (0x04)	Reached start of a track	O
EVENT_PLAYBACK_POS_CHANGED (0x05)	Change in playback position. Returned after the specified playback notification change notification interval	O
EVENT_BATT_STATUS_CHANGED (0x06)	Change in battery status	O
EVENT_SYSTEM_STATUS_CHANGED (0x07)	Change in system status	O
EVENT_PLAYER_APPLICATION__SETTING_CHANGED (0x08)	Change in player application setting	O
0x09-0xFF	Reserved for future use	

24 Appendix I: Examples of PDUs for different command and responses

Get Capability command for Company ID

Oct	**MSB (7)**	**6**	**5**	**4**	**3**	**2**	**1**	**LSB (0)**
0	0x0				Ctype: 0x1 (STATUS)			
1	Subunit_type:0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x0 (VENDOR DEPENDENT)							
3 -5	Company ID: BT SIG registered CompanyID							
6	PDU ID (0x10 - Get Capabilities)							
7	Reserved (0x00)						Packet Type (0x0)	
8 – 9	Parameter Length (0x0001)							
10	Capability ID: 0x2 (CompanyID)							

Get Capability response for Company ID

Oct	**MSB (7)**	**6**	**5**	**4**	**3**	**2**	**1**	**LSB (0)**
0	0x0				Response: 0xC (STABLE)			
1	Subunit_type: 0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x0 (VENDOR DEPENDENT)							
3 -5	Company ID: BT SIG registered CompanyID							
6	PDU ID: 0x10 (Get Capabilities)							
7	Reserved: 0x00						Packet Type: 0x0	
8 – 9	Parameter Length: 0x5							
10	Capability ID: 0x2 (CompanyID)							
11	Capability Count: 0x1							
12- 14	Company ID: BT SIG registered CompanyID							

Get Capability command for Events

Oct	**MSB (7)**	**6**	**5**	**4**	**3**	**2**	**1**	**LSB (0)**
0	0x0				Ctype: 0x1 (STATUS)			
1	Subunit_type:0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x0 (VENDOR DEPENDENT)							
3 -5	Company ID: BT SIG registered CompanyID							
6	PDU ID (0x10 - Get Capabilities)							
7	Reserved (0x00)						Packet Type (0x0)	
8 – 9	Parameter Length (0x0001)							
10	Capability ID: 0x3 (EventsID)							

Get Capability response for Events

Oct	**MSB (7)**	**6**	**5**	**4**	**3**	**2**	**1**	**LSB (0)**
0	0x0				Response: 0xC (STABLE)			
1	Subunit_type: 0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x0 (VENDOR DEPENDENT)							
3 -5	Company ID: BT SIG registered CompanyID							
6	PDU ID: 0x10 (Get Capabilities)							
7	Reserved: 0x00						Packet Type: 0x0	
8 – 9	Parameter Length: 0x5							
10	Capability ID: 0x3 (EventsID)							
11	Capability Count: 0x3							
12	EventID1: 0x1 (EVENT_PLAYBACK_STATUS_CHANGED)							
13	EventID2: 0x2 (EVENT_TRACK_CHANGED)							
14	EventID3: 0x8 (EVENT_PLAYER_APPLICATION_SETTING_CHANGED)							

List Application Settings Attributes command

Oct	**MSB (7)**	**6**	**5**	**4**	**3**	**2**	**1**	**LSB (0)**
0	0x0				Ctype: 0x1 (STATUS)			
1	Subunit_type:0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x0 (VENDOR DEPENDENT)							
3 -5	Company ID: BT SIG registered CompanyID							
6	PDU ID (0x11 – ListApplicationSettingAttributes)							
7	Reserved (0x00)						Packet Type (0x0)	
8 – 9	Parameter Length (0x0)							

List Application Settings Attributes response

Oct	**MSB (7)**	**6**	**5**	**4**	**3**	**2**	**1**	**LSB (0)**
0	0x0				Response: 0xC (STABLE)			
1	Subunit_type: 0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x0 (VENDOR DEPENDENT)							
3 -5	Company ID: BT SIG registered CompanyID							
6	PDU ID (0x11 – ListApplicationSettingAttributes)							
7	Reserved: 0x00						Packet Type: 0x0	
8 – 9	Parameter Length: 0x3							
10	NumPlayerApplicationSettingAttributes: 0x2							
11	PlayerApplicationSettingAttributeID1: 0x1 (Equalizer ON/OFF Status)							
12	PlayerApplicationSettingAttributeID2: 0x3 (Shuffle ON/OFF Status)							

Registration for notification of Event Track changed

Oct	**MSB (7)**	**6**	**5**	**4**	**3**	**2**	**1**	**LSB (0)**
0	0x0				Ctype: 0x3 (NOTIFY)			
1	Subunit_type:0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x0 (VENDOR DEPENDENT)							
3 -5	Company ID: BT SIG registered CompanyID							
6	PDU ID (0x31 – Register Notification)							
7	Reserved (0x00)						Packet Type (0x0)	
8 – 9	Parameter Length (0x5)							
10	EventID2: 0x2 (EVENT_TRACK_CHANGED)							
11 14	Playback interval: 0x0 (*Ignored for this event*)							

Register Notification interim response

Oct	**MSB (7)**	**6**	**5**	**4**	**3**	**2**	**1**	**LSB (0)**
0	0x0				Response: 0xF (INTERIM)			
1	Subunit_type: 0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x0 (VENDOR DEPENDENT)							
3 -5	Company ID: BT SIG registered CompanyID							
6	PDU ID: 0x31 (Register Notification)							
7	Reserved: 0x00						Packet Type: 0x0	
8 – 9	Parameter Length: 0x9							
10	EventID2: 0x2 (EVENT_TRACK_CHANGED)							
11- 18	Identifier: 0xFFFFFFFF							

Register Notification response

Oct	MSB (7)	6	5	4	3	2	1	LSB (0)
0	0x0				Response: 0xD (CHANGED)			
1	Subunit_type: 0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x0 (VENDOR DEPENDENT)							
3 -5	Company ID: BT SIG registered CompanyID							
6	PDU ID: 0x31 (Register Notification)							
7	Reserved: 0x00						Packet Type: 0x0	
8 – 9	Parameter Length: 0x9							
10	EventID2: 0x2 (EVENT_TRACK_CHANGED)							
11- 18	Identifier: 0x8							

Get Element Attributes command

Oct	MSB (7)	6	5	4	3	2	1	LSB (0)
0	0x0				Ctype: 0x1 (STATUS)			
1	Subunit_type:0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x0 (VENDOR DEPENDENT)							
3 -5	Company ID: BT SIG registered CompanyID							
6	PDU ID (0x20 – GetElementAttributes)							
7	Reserved (0x00)						Packet Type (0x0)	
8 – 9	Parameter Length (0xD)							
10-13	Identifier: 0x0 (PLAYING)							
14	AttributeCount: 0x2							
15-22	Attribute1: 0x1 (TitleOfMedia)							
23-26	Attribute2: 0x7 (Playing Time)							

Get Element Attributes response

Oct	MSB (7)	6	5	4	3	2	1	LSB (0)
0	0x0				Response: 0xC (STABLE)			
1	Subunit_type:0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x0 (VENDOR DEPENDENT)							
3 -5	Company ID: BT SIG registered CompanyID							
6	PDU ID (0x20 – GetElementAttributes)							
7	Reserved (0x00)						Packet Type (0x0)	
8– 9	Parameter Length (0x2A)							
11	Number of Attributes (0x2)							
12-19	Attribute ID 1: 0x1 (TitleOfMedia)							
20-21	CharacterSetID1: 0x6A (UTF-8)							
22-23	AttributeValueLength1: 0x13							
24-42	AttributeValue1: ‘Give Peace a Chance’							
43-50	Attribute ID 2: 0x7 (Playing Time)							
51-52	CharacterSetID2: 0x6A (UTF-8)							
53-54	AttributeValueLength2: 0x6							

55-57	AttributeValue2: ‘103000’ (= 103000 ms – 103 sec. – 1min43s)

PASS THROUGH Command (vendor unique) for Group Navigation

Oct	MSB (7)	6	5	4	3	2	1	LSB (0)
0	0x0				Ctype: 0x0 (CONTROL)			
1	Subunit_type:0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x7C (PASS THROUGH)							
3	State_flag *2	Operation_ID: 0x7E (VENDOR UNIQUE)						
4	Operation_data_field_length: 0x5							
5 - 7	Company ID: BT SIG registered CompanyID							
8 - 9	Vendor_unique_id							

PASS THROUGH Response (vendor unique) for Group Navigation

Oct	MSB (7)	6	5	4	3	2	1	LSB (0)
0	0x0				Response *1			
1	Subunit_type:0x9 (PANEL)					Subunit_ID: 0x0		
2	Opcode: 0x7C (PASS THROUGH)							
3	State_flag *2	Operation_ID: 0x7E (VENDOR UNIQUE)						
4	Operation_data_field_length: 0x5							
5 - 7	Company ID: BT SIG registered CompanyID							
8 - 9	Vendor_unique_id							

*1 0x8(NOT_IMPLEMENTED), 0x9 (ACCEPTED), 0xA (REJECTED)

*2 A CT sends a command frame with its state_flag field in value 0 when a button is pushed and in value 1 when the button is released.

25 Appendix J: List of Example MSC of different Metadata Transfer Commands

25.1 InformDisplayableCharacterSet

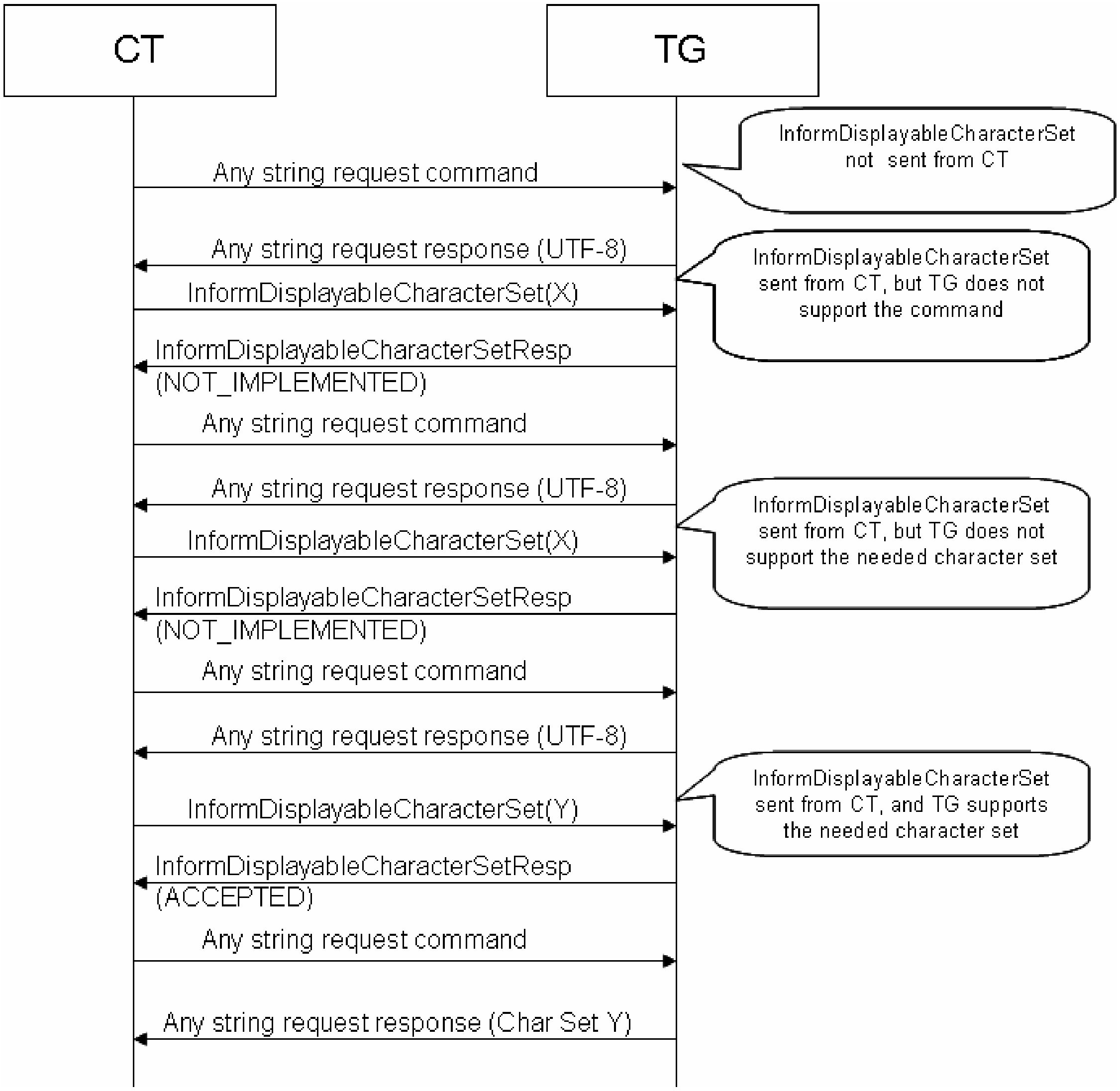

CT
TG
InformDisplayableCharacterSet not sent from CT
Any string request command
Any string request response (UTF-8)
InformDisplayableCharacterSet sent from CT, but TG does not support the command
InformDisplayableCharacterSet(X)
InformDisplayableCharacterSetResp (NOT_IMPLEMENTED)
Any string request command
Any string request response (UTF-8)
InformDisplayableCharacterSet sent from CT, but TG does not support the needed character set
InformDisplayableCharacterSet(X)
InformDisplayableCharacterSetResp (NOT_IMPLEMENTED)
Any string request command
Any string request response (UTF-8)
InformDisplayableCharacterSet sent from CT, and TG supports the needed character set
InformDisplayableCharacterSet(Y)
InformDisplayableCharacterSetResp (ACCEPTED)
Any string request command
Any string request response (Char Set Y)

Figure 25.1: Example of using InformDisplayableCharacterSet

25.2 RegisterNotification

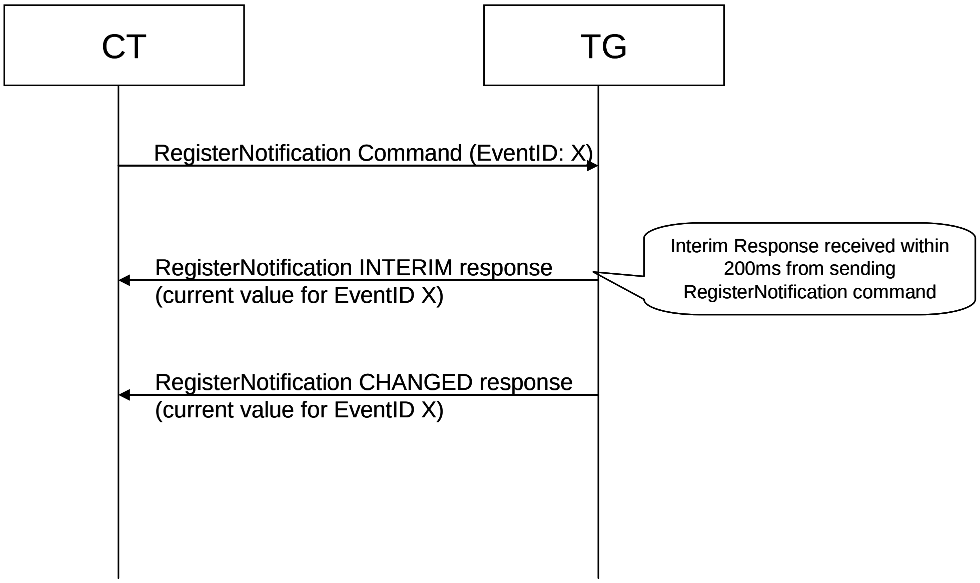

CT
TG
RegisterNotification Command (EventID: X)
RegisterNotification INTERIM response
(current value for EventID X)
Interim Response received within 200ms from sending RegisterNotification command
RegisterNotification CHANGED response
(current value for EventID X)

Figure 25.2: Example of using RegisterNotification

25.3 RequestContinuingResponse

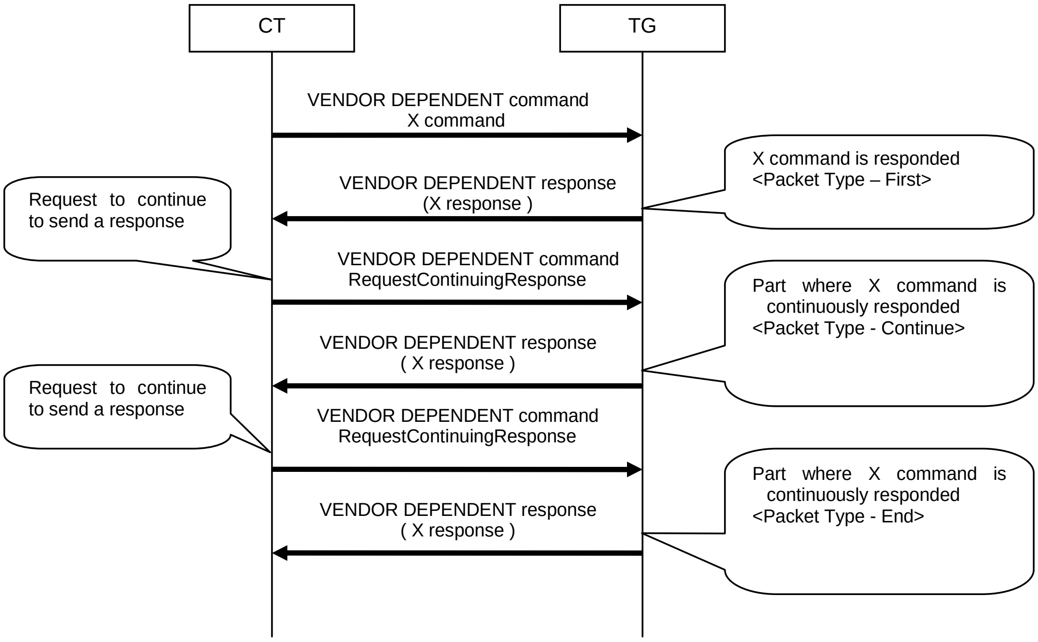

CT
TG
VENDOR DEPENDENT command
X command
X command is responded
<Packet Type – First>
VENDOR DEPENDENT response
(X response)
Request to continue to send a response
VENDOR DEPENDENT command
RequestContinuingResponse
Part where X command is continuously responded
<Packet Type - Continue>
VENDOR DEPENDENT response
(X response)
Request to continue to send a response
VENDOR DEPENDENT command
RequestContinuingResponse
Part where X command is continuously responded
<Packet Type - End>
VENDOR DEPENDENT response
(X response)

Figure 25.3: Example of using RequestContinuingResponse

25.4 AbortContinuingResponse

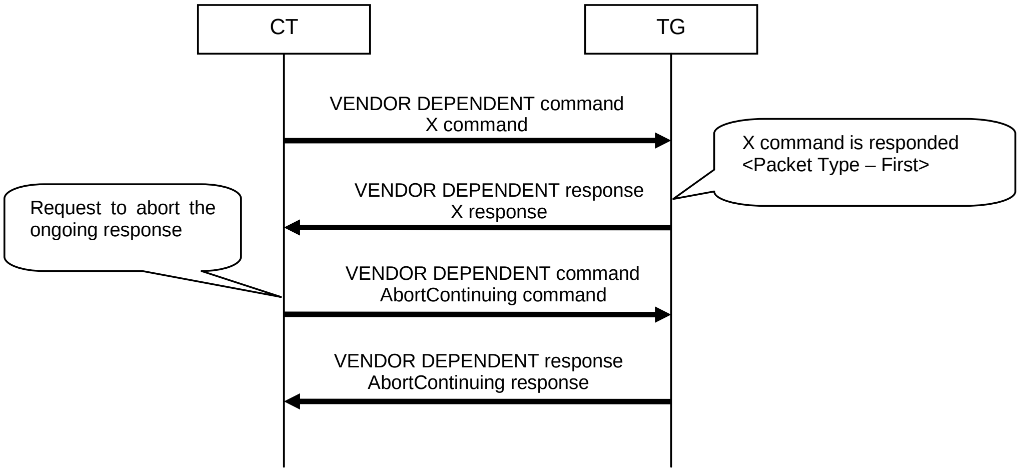


Figure 25.4: Example of using AbortContinuingResponse

26 Appendix K: Acronyms and Abbreviations

Acronym	Description
1394TA	1394 Trade Association
A/V	Audio/Video
AV/C	The AV/C Digital Interface Command Set
AVCTP	Audio/Video Control Transport Protocol
AVRCP	Audio/Video Remote Control Profile
CT	Controller
ICS	Implementation Conformance Statement
IEEE	The Institute of Electrical and Electronics Engineers
LC	Link Controller
LM	Link Manager
MTU	Maximum Transmission Unit
PSM	Protocol/Service Multiplexer
QoS	Quality of Service
RFA	Reserved for Future Additions
RFD	Reserved for Future Definition
SDP	Service Discovery Protocol
TG	Target
TP	Test Purpose
TSS	Test Suite Structure