DLNA Guidelines
December 2011

Part 1: Architectures and Protocols

An Industry Guide for
Building Interoperable
Platforms, Devices,
and Applications

Fulfilling the promise of the digital home requires a cross-industry effort to develop and promote a common industry framework for interoperability. This industry framework is expressed through the DLNA Guidelines document that has been developed to provide Consumer Electronic, Mobile Device and PC companies with the information needed to build interoperable platforms, devices, and application for the digital home.

Do Not Copy

Legal Disclaimer

CONTENTS

Introduction

Consumers are acquiring, viewing, and managing an increasing amount of digital media (photos, music, and video) on devices in the Consumer Electronics (CE), Mobile Device, and Personal Computer (PC) domains. Consumers want to conveniently enjoy that content-regardless of the source-across different devices and locations in their homes. The digital home vision integrates the Internet, mobile, and broadcast networks through a seamless, interoperable network, which will provide a unique opportunity for manufacturers and consumers alike. In order to deliver on this vision, it was recognized that a common set of industry design guidelines would be required to allow companies to participate in a growing marketplace, leading to more innovation, simplicity, and value for consumers.

The Digital Living Network Alliance answered this challenge by taking the initiative to develop a workable framework for interoperable product design. The DLNA Home Networked Device Interoperability Guidelines has been created in a unique cross-indust ry effort that combined the efforts of over 100 Consumer Electronics, PC-industry and Mobile Device companies from around the world who worked together with the aim of achieving the world's first substantial platform for true interoperability between personal computer and consumer electronic devices. The Interoperability Guidelines provide product developers with a long-term architectural view, plus specific guidance for IP-networked platforms, devices and applications in the home. The Interoperability Guidelines will be introduced in phases over several years to accompany the market adoption of usages and the availability of needed technology and standards.

1 Scope

1.1 Purpose

The Interoperability Guidelines consists of five parts covering Architecture and Protocols, Media Formats, Link Protection, DRM Interoperability Systems and Device Profiles. It provides vendors with the information needed to build interoperable networked platforms and devices for the digital home. The necessary standards and technologies are available now to enable products to be built for networked entertainment centric usages. However, standards and technologies need to be clarified and options limited to ensure interoperability. The DLNA Home Networked Device Interoperability Guidelines fulfill that role.

The Interoperability Guidelines are based on an architecture (see Clause 4) that defines interoperable components for devices and software infrastructure. It covers physical media, network transports, device discovery and control, media management and control, media formats, media transport protocols, and remote user interfaces. Table 1 shows a summary of the key functional components and technology ingredients that are covered by the Interoperability Guidelines.

Table 1 — Key Technology Ingredients

Functional Components	Technology Ingredients
Connectivity	Ethernet*, 802.11, MoCA, HPNA and Bluetooth
Networking	IPv4 Suite
Device Discovery and Control	UPnP* Device Architecture v1.0
Media Management and Control	UPnP AV and UPnP Printer:1
Media Formats	Required and Optional Format Profiles

Functional Components	Technology Ingredients
Media Transport	HTTP (Mandatory) and RTP (Optional)
Remote User Interfaces	CEA-2014-A

1.2 Version Number

This document specifies the core architecture and protocols of DLNA implementations. For version control, the protocols defined in this document constitute version 1.5 of the specifications. Device implementations advertise adherence to the protocols selecting value 1.5 in the fields and flags designed to expose the DLNA protocol version.

1.3 Audience

The Interoperability Guidelines are intended for the following audiences:

- Marketing professionals who specify requirements for home networked media products.
- Developers who design and build home networked media products.
- Quality assurance personnel who test and validate home networked media products.

1.4 Organization

This part of the Interoperability Guidelines is organized as follows:

Clause 2 Normative References – Acquisition information on all normative references contained in this document.

Clause 3 Terms, definitions, symbols and abbreviated terms – Definitions of abbreviations and terms used in this document.

Clause 4 DLNA Home Network Architecture – An overview of the DLNA home networking architecture.

Clause 5 DLNA Device Model – An overview of the major device categories used to group guideline requirements.

Clause 6 Guideline Terminology and Conventions – Definitions for the compliance and usage classifications used for guideline requirements.

Clause 7 Guideline Requirements – Covers guideline requirements for DLNA devices excluding Media Formats and Common Link Protection which are covered in other parts.

Annex A – Covers a set of recommendations for home network infrastructure devices such as gateways, routers, and hubs to ensure they work well with DLNA devices.

Annex B – Describes the way DLNA devices should represent tuner-based content.

Annex C – Describes how a DLNA device can represent itself on multiple network interfaces. The annex also discusses how a Content Source should expose content URI values for different network interfaces.

Annex D – Introduces developers to the technical considerations required to support printers and also discusses some of the usability aspects of printing that are important for a good user experience.

Annex E – Clarifies the general applicability of the Uniform Client Data Availability Model (UCDAM). It describes the data accessibility assumptions for both Content Sources and Content Receivers. The UCDAM model strives for completeness by using examples derived from stored, converted, and live content streams. The model also accounts for caching of data by Content Receivers.

Annex F – Provides guidance for developers on extending Auto-IP support for IP Stacks that have problems with full conformance to Auto-IP.

Annex G – Provides guidance on network connectivity for mobile devices, including Bluetooth security and NC power saving modes.

Annex H – Provides graphic layout of the protocol stack for the RTP transport and SDP/RTSP/RTCP parameters.

Annex I — Clarifies implementations of IP address conflict resolution in Auto-IP.

Annex J — Provides guidance on using Wi-Fi Direct with DLNA System Usages

Annex K — Provides theory of operation for the EPG System Usage

Annex L — Provides EPG rating systems used in DLNA

Annex M — Acquisition information on all informative references contained in this document.

2 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

[1] IEEE 802.1D-2004, Annex G, IEEE Standard for Information technology - Telecommunications and information exchange between systems - IEEE standard for local and metropolitan area networks - Common specifications - Media access control (MAC) Bridges, June 9, 2004.
http://standards.ieee.org/getieee802/index.html

[2] IEEE 802.1Q-2003, IEEE Standard for Information Technology - Telecommunications and information exchange between systems - IEEE standard for local and metropolitan area networks - Common specifications - Virtual Bridged Local Area Networks, May 7, 2003.
http://standards.ieee.org/getieee802/index.html

[3] IEEE 802.3-2002, IEEE Standard for information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specification, March 8, 2002.
http://standards.ieee.org/getieee802/index.html

[4] ISO 8601:2004, Data elements and interchange formats – Information interchange -- Representation of dates and times, International Standards Organization, December 3, 2004.
http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40874&ICS1=1&ICS2=140&ICS3=30

[5] ISO/IEC 13818-1:2000, Information technology -- Generic coding of moving pictures and associated audio information: Systems, International Standards Organization, 2000.
http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=31537

[6] ISO/IEC 13818-9:1996, Information technology -- Generic coding of moving pictures and associated audio information -- Part 9: Extension for real time interface for systems decoders, International Standards Organization, 1996.
http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=25434&ICS1=35&ICS2=40&ICS3=

[7] IETF RFC 3927, Dynamic Configuration of IPv4 Link-Local addresses, Stuart Cheshire, Apple Computer, B. Aboba, Microsoft Corporation, E.Guttman, Sun Microsystems, May 2005.
http://www.ietf.org/rfc/rfc3927.txt

[8] IETF RFC 4585 Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-based Feedback (RTP/AVPF), Joerg Ott, Uni Bremen TZI, Stephan Wenger, TU Berlin, Noriyuki Sato, Oki, Carsten Burmeister, Matsushita, Joe Rey, Matsushita, July 2006.
http://www.ietf.org/rfc/rfc4585.txt

[9] IETF RFC 4184, RTP Payload Format for AC-3 Audio, B. Link T. Hager, Dolby Laboratories, J. Flanks, Microsoft, October 2005.
http://www.ietf.org/rfc/rfc4184.txt

[10] IETF RFC 4588, RTP Retransmission Payload Format, J. Rey, Panasonic, D. Leon, Nokia, A. Miyazaki, Panasonic, V. Varsa, Nokia, R. Hakenberg, Panasonic, July 2006.
http://www.ietf.org/rfc/rfc4588.txt

[11] IETF RFC 768, User Datagram Protocol, J. Postel, August 28, 1980.
http://www.ietf.org/rfc/rfc0768.txt

[12] IETF RFC 791, Internet Protocol, J. Postel, September 1981.
http://www.ietf.org/rfc/rfc0791.txt

[13] IETF RFC 792, Internet Control Message Protocol, J. Postel, September 1981.
http://www.ietf.org/rfc/rfc0792.txt

[14] IETF RFC 793, Transmission Control Protocol, J. Postel, September 1981.
http://www.ietf.org/rfc/rfc0793.txt

[15] IETF RFC 826, An Ethernet Address Resolution Protocol - or - Converting Network Protocol Addresses to 48.bit Ethernet Addresses for Transmission on Ethernet Hardware, David C. Plummer, November 1982.
http://www.ietf.org/rfc/rfc0826.txt

[16] IETF RFC 1122, Requirements for Internet Hosts - Communications Layers, R. Braden, October 1989.
http://www.ietf.org/rfc/rfc1122.txt

[17] IETF RFC 1191, Path MTU Discovery, J. Mogul, DECWRL, S. Deering, Stanford University, November 1990.
http://www.ietf.org/rfc/rfc1191.txt

[18] IETF RFC 1305, Network Time Protocol (Version 3), Specification, Implementation and Analysis, David L. Mills, University of Delaware, March 1992.
http://www.ietf.org/rfc/rfc1305.txt

[19] IETF RFC 1738, Uniform Resource Locators (URL), T. Berners-Lee, CERN, L. Masinter, Xerox Corporation, M. McCahill, University of Minnesota, December 1994.
http://www.ietf.org/rfc/rfc1738.txt

[20] IETF RFC 1812, Requirements for IP version 4 Routers, F. Baker, June 1995.
http://www.ietf.org/rfc/rfc1812.txt

[21] IETF RFC 1945, Hypertext Transfer Protocol - HTTP/1.0, T. Berners-Lee, MIT/LCS, R. Fielding, UC Irvine, H. Frystyk, May 1996.
http://www.ietf.org/rfc/rfc1945.txt

[22] IETF RFC 2131, Dynamic Host Configuration Protocol, R. Droms, March 1997.
http://www.ietf.org/rfc/rfc2131.txt

[23] IETF RFC 2145, Use and Interpretation of HTTP Version Numbers, J. C. Mogul, DEC, R. Fielding, UC Irvine, J. Gettys, DEC, H. Frystyk, MIT/LCS, May 1997.
http://www.ietf.org/rfc/rfc2145.txt

[24] IETF RFC 2234, Augmented BNF for Syntax Specifications: ABNF, Ed D. Crocker, Internet Mail Consortium, P. Overell, Demon Internet Ltd., November 1997.
http://www.ietf.org/rfc/rfc2234.txt

[25] IETF RFC 2250, RTP Payload Format for MPEG1/MPEG2 Video, D. Hoffman, G. Fernando, Sun Microsystems, Inc., V. Goyal, Precept Software, Inc. M. Civanlar, AT&T Labs - Research, January 1998.
http://www.ietf.org/rfc/rfc2250.txt

[26] IETF RFC 2279, UTF-8, a transformation format of ISO 10646, F. Yergeau, Alis Technologies, January 1998.
http://www.ietf.org/rfc/rfc2279.txt

[27] IETF RFC 2326, Real Time Streaming Protocol (RTSP), H. Schulzrinne, Columbia U., A. Rao, Netscape, R. Lanphier, RealNetworks, April 1998.
http://www.ietf.org/rfc/rfc2326.txt

[28] IETF RFC 2327, SDP: Session Description Protocol, M. Handley, V. Jacobson, ISI/LBNL, April 1998.
http://www.ietf.org/rfc/rfc0791.txt

[29] IETF RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, MIT/LCS, R. Fielding, U.C. Irvine, L. Masinter, Xerox Corporation, August 1998.
http://www.ietf.org/rfc/rfc2396.txt

[30] IETF RFC 2429, RTP Payload Format for the 1988 Version of ITU-T Rec. H.263 Video (H.263+), December 2004.
http://www.ietf.org/rfc/rfc2429.txt

[31] IETF RFC 4352, RTP Payload Format for the Extended Adaptive Multi-Rate Wideband (AMR-WB+) Audio Codec, Johan Sjoberg, Magnus Westerlund, Ericsson, Ari Lakaniemi, Stephan Wenger, Nokia, January 2006.
http://www.ietf.org/rfc/rfc4352.txt

[32] IETF RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, K. Nichols, Cisco Systems, S. Blake, Torrent Networking Technologies, F. Baker, Cisco Systems, D.Black, EMC Corporation, December 1998.
http://www.ietf.org/rfc/rfc2474.txt

[33] IETF RFC 2616, Hypertext Transfer Protocol - HTTP/1.1, R. Fielding, UC Irvine, J. Gettys, Compaq/W3C, J. Mogul, Compaq, H. Frystyk, W3C/MIT, L. Masinter, Xerox, P. Leach, Microsoft, T. Berners-Lee, June 1999.
http://www.ietf.org/rfc/rfc2616.txt

[34] IETF RFC 2822, Internet Message Format, P. Resnick, QUALCOMM Incorporated, April 2001.
http://www.ietf.org/rfc/rfc2822.txt

[35] IETF RFC 3261, SIP: Session Initiation Protocol, J. Rosenberg, dynamicsoft, H. Schulzrinne, Columbia U., G. Camarillo, Ericsson, A. Johnston, Worldcom, J. Peterson, Neustar, R. Sparks, dynamicsoft, M. Handley, ICIR, E. Schooler, AT&T, June 2002.
http://www.ietf.org/rfc/rfc3261.txt

[36] IETF RFC 3267, Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs, J. Sjoberg, M. Westerlund, Ericsson, A. Lakaniemi, Nokia, Q. Xie, Motorola, June 2002.
http://www.ietf.org/rfc/rfc3267.txt

[37] IETF RFC 3391, The MIME Application/Vnd.pwg-multiplexed Content-Type, R. Herriot, December 2002.
http://www.ietf.org/rfc/rfc3391.txt

[38] IETF RFC 3550, RTP: A Transport Protocol for Real-Time Applications, H. Schulzrinne, Columbia University, S. Casner, Packet Design, R. Frederick, Blue Coat Systems Inc., V. Jacobson, Packet Design, July 2003.
http://www.ietf.org/rfc/rfc3550.txt

[39] IETF RFC 3551, RTP Profile for Audio and Video Conferences with Minimal Control, H. Schulzrinne, Columbia University, S. Casner, Packet Design, July 2003.
http://www.ietf.org/rfc/rfc3551.txt

[40] IETF RFC 3555, MIME Type Registration of RTP Payload Formats, S. Casner, Packet Design, P. Hoschka, W3C/INRIA/MIT, July 2003.
http://www.ietf.org/rfc/rfc3555.txt

[41] IETF RFC 3556, Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth, S. Casner, Packet Design, July 2003.
http://www.ietf.org/rfc/rfc3556.txt

[42] IETF RFC 3640, RTP Payload Format for Transport of MPEG-4 Elementary Streams, J. van der Meer, Philips Electronics, D. Mackie, Apple Computer, V. Swaminathan, Sun Microsystems Inc., D. Singer, Apple Computer, P. Gentric, Philips Electronics, November 2003.
http://www.ietf.org/rfc/rfc3640.txt

[43] IETF RFC 3984, RTP Payload Format for H.264 Video, S. Wenger, M. M. Hannuksela, T. Stockhammer, M. Westerlund, D. Singer, January 2005.
http://www.ietf.org/rfc/rfc3984.txt

[44] ISO/IEC 29341-3-10, AVv1 – Information Technology – UPnP Device Architecture – Part 3-10: Audio Video Device Control Protocol – Audio Video Transport Service

ISO/IEC 29341-4-10, AVv2, AVv3 – Information Technology - UPnP Device Architecture - Part 4-10: Audio Video Device Control Protocol - Level 2 - Audio Video Transport Service

[45] ISO/IEC 29341-3-11, AVv1 – Information Technology – UPnP Device Architecture – Part 3-11: Audio Video Device Control Protocol – Connection Manager Service

ISO/IEC 29341-4-11, AVv2, AVv3 – Information Technology – UPnP Device Architecture – Part 4-11: Audio Video Device Control Protocol – Level 2 – Connection Manager Service

[46] ISO/IEC 29341-3-12:2008, AVv1 – Information Technology – UPnP Device Architecture – Part 3-12: Audio Video Device Control Protocol – Content Directory Service

ISO/IEC 29341-4-12, AVv2 – Information Technology - UPnP Device Architecture - Part 4-12: Audio Video Device Control Protocol - Level 2 - Content Directory Service

ISO/IEC 29341-14-12, AVv3 – Information Technology – UPnP Device Architecture – Part 14-12: Audio Video Device Control Protocol – Level 3 – Content Directory Service

[47] ISO/IEC 29341-1, Information Technology - UPnP Device Architecture - Part 1-1: UPnP Device Architecture Version 1.0

[48] ISO/IEC 29341-3-2, AVv1 – Information Technology - UPnP Device Architecture - Part 3-2: Audio Video Device Control Protocol - Media Renderer Device

ISO/IEC 29341-4-2, AVv2, AVv3 – Information Technology - UPnP Device Architecture - Part 4-2: Audio Video Device Control Protocol - Level 2 - Media Renderer Device

[49] ISO/IEC 29341-3-3, AVv1 – Information Technology - UPnP Device Architecture - Part 3-3: Audio Video Device Control Protocol - Media Server Device

ISO/IEC 29341-4-3, AVv2 – Information Technology - UPnP Device Architecture - Part 4-3: Audio Video Device Control Protocol - Level 2 - Media Server Device

ISO/IEC 29341-14-3, AVv3 – Information Technology – UPnP Device Architecture – Part 14-3: Audio Video Device Control Protocol – Level 3 – Media Server Device

[50] ISO/IEC 29341-9-12, Information Technology - UPnP Device Architecture - Part 9-12: Imaging Device Control Protocol - Print Basic Service

[51] ISO/IEC 29341-3-13, AVv1 – Information Technology - UPnP Device Architecture - Part 3-13: Audio Video Device Control Protocol - Rendering Control Service

ISO/IEC 29341-4-13, AVv2, AVv3 – Information Technology - UPnP Device Architecture - Part 4-13: Audio Video Device Control Protocol - Level 2 - Rendering Control Service

[52] CSS Print Profile, CSS Print Profile, Jim Bigelow (editor), W3C, February 25, 2004.
http://www.w3.org/TR/css-print

[53] XHTML-Print, XHTML-Print, Jim Bigelow (editor), W3C, January 31, 2006.
http://www.w3.org/TR/xhtml-print/

[54] XHTML Photo Templates, XHTML-Print Photo Templates for UPnP PrintEnhanced:1 v1.0, UPnP Forum, May 4, 2005.
http://www.upnp.org/standardizeddcps/documents/Phototemplates_v1_050504.pdf

[55] XML W3C Recommendations, Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation, February 4, 2004.
http://www.w3.org/TR/REC-xml

[56] IEC 62481-2:2012 Digital Living Network Alliance (DLNA) Guidelines - Part 2: Media format profiles

[57] IEC 62481-3:2012 Digital Living Network Alliance (DLNA) Guidelines – Part 3: Link protection

[58] IEC 60169-24: Radio-frequency coaxial connectors with screw coupling, typically for use in 75 ohm cable distribution systems (Type F)
http://webstore.iec.ch/webstore/webstore.nsf/artnum/001029

[59] ITU-T, Recommendation G.9954 (01/07), Home networking transceivers – Enhanced physical, media access, and link layer specifications. ITU-T SG15/Q4 T-REC-G.9954-200701-I, Geneva, Switzerland, (01/2007)
http://www.itu.int/rec/T-REC-G.9954-200701-I
http://www.homepna.org

[60] ISO/IEC 29341-4-14, AVv2, AVv3 – Information Technology - UPnP Device Architecture - Part 4-14: Audio Video Device Control Protocol - Level 2 - Scheduled Recording Service

[61] ISO/IEC 29341-4-4, AVv2, AVv3 – Information Technology - UPnP Device Architecture - Part 4-4: Audio Video Device Control Protocol - Level 2 - Audio Video Data Structures

[62] ETSI TS 102 822-3, Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"), Part 3: Metadata

[63] ETSI EN 300 468: Specification for Service Information (SI) in DVB systems, Digital Video Broadcasting (DVB)

[64] IETF RFC 3066 Tags for the Identification of Languages, January 2001

[65] ETF RFC 4646 Tags for the Identification of Languages, January 2006

[66] ISO/IEC 14977:1996, EBNF, Information technology -- Syntactic metalanguage -- Extended BNF

[67] ETSI TS 102 822-4 Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 4: Content referencing

[68] ISO 3166 Codes for the representation of names of countries and their subdivisions

[69] W3C Namespaces in XML 1.0 (Third Edition) http://www.w3.org/TR/xml-names/

[70] W3C XQuery 1.0 and XPath 2.0 Functions and Operators (Second Edition) http://www.w3.org/TR/2010/REC-xpath-functions-20101214/

3 Terms, definitions, symbols and abbreviated terms

3.1 Definition of Terms

For the purposes of this document, the following terms and definitions apply.

3.1.1
Authentication
The process by which an entity verifies that another entity is who or what it claims to be. (Examples of entities are: device, person, process.)

Authentication for Bluetooth is defined in [82] as the process of verifying which device is at the other end of the link. Authentication is performed for devices based on their Bluetooth Address (BD_ADDR). In Bluetooth this is achieved by the authentication procedure based on the mutually stored link key or by pairing (See below)

3.1.2
Authorization
The process of access check to determine whether the authenticated entity has access to the resource.

Authorization for Bluetooth is defined in [82] as the process of deciding if a device is allowed to have access to a particular service. This is where the concept of 'trusted' exists. Trusted devices (authenticated and indicated as "trusted"), are allowed access to services. Untrusted or unknown devices might require authorization based on user interaction before access to services is granted. This does not exclude the case where the authorization might be given by an application automatically. Authorization for Bluetooth always includes authentication.

3.1.3
Background Transfer
A media transfer mode where the target content binary is transferred at any rate up to the maximum achievable rate for the communication channel and the two communicating parties. This mode is typically used for example when downloading or uploading content.

3.1.4
Basic Tuner
A DLNA tuner implementation as defined in the initial version of the DLNA Interoperability Guidelines.

3.1.5
Bearer
A physical and link-level network transport, such as Bluetooth or 802.11.

3.1.6
Byte-based Seek Operations
This process identifies the Rendering Endpoint request and the Content Source response that allows the former to obtain a segment of the content for playback, where such a segment is specified in units of bytes.

3.1.7
Cacheable Content
ontent binaries whose binary representations remain static over time are considered cacheable content. By default HTTP allows intermediate HTTP caches to store such items and respond to similar HTTP requests as a means to accelerate the interaction with the user. In these Interoperability Guidelines cacheable content includes (but it is not limited to):

- Image, Audio, AV content that exists as stored files.
- Content resulting from transcoding or encoding operations when the output binaries can be preserved over time.

3.1.8
Channel
A channel refers to one or more media streams that together constitute a unique entity for the purpose of announcement, selection, and rendering. For example, for digital television sources, a channel is equivalent to an ATSC "virtual channel", a DVB "service", or an MPEG-2 "Program". For digital radio sources, a channel is equivalent to a single "station".

3.1.9
Channel Lineup Container
Available tuner channels exposed in a CDS container for an Extended Tuner. The content may or may not be streamed over the network.

3.1.10
Channelized Content
Content that is available via a particular distribution channel such as a tuner and is available on that channel according to a schedule.

3.1.11
Content
An aggregation of one or more works.

3.1.12
Content Binary
A binary representation of content for the purpose of storage or transfer over communication links.

3.1.13
Content Transformation
Transcoding, transrating, or scaling of a content binary.

3.1.14
Content Source
The endpoint that places content onto the network for transfer to another endpoint.

3.1.15
Content Receiver
The endpoint that consumes content received via a network transfer from another endpoint.

3.1.16
Controller-byte Seek Operations
This process identifies the Control Point request and the UPnP AV MediaRenderer response that allows the former to specify a segment of the content to be rendered by the latter, where such a segment is specified in units of bytes.

3.1.17
Controller-time Seek Operations
This process identifies the Control Point request and the UPnP AV MediaRenderer response that allows the former to specify a segment of the content to be rendered by the latter, where such a segment is specified in units of time.

3.1.18
Decoder Friendly Point
A decoder friendly point is a point in the stream that the decoder can begin to process data without any other internal state information about the stream. The decoder can begin processing at that point and create a valid output rendering.

3.1.19
Device Capability
A set of Device Functions (at least 1) aggregated to support a System Usage. A Device Capability cannot stand alone, and must be deployed in conjunction with an implementation of a valid DLNA Device Class. Since a Device Capability does not stand alone, it is not required to have components in all layers of the DLNA architecture. A Device Capability can have a one to one correspondence to a Device Function. A Device Capability is a certifiable entity only when it is implemented as an addition to at least one Device Class.

3.1.20
Device Category
A Device Category is a group of Device Classes with the same environmental characteristics and sharing common System Usages that are enabling home networking use case scenarios. Examples used within this document are HND (Home Network Device), MHD (Mobile Handheld Device), and HID (Home Infrastructure Device). While Device Classes are grouped within a Device Category, a single physical device can support Device Classes that fall into multiple Device Categories.

3.1.21
Device Class
A Device Class is defined by a set of Device Functions. It specifies the features supported on a device regardless of its physical attributes. Examples used within this document are DMS (Digital Media Server) and DMP (Digital Media Player). A single device can support multiple Device Classes. A DLNA device must support a least one Device Class and can support one or more Device Capabilities A Device Class is the certifiable entity in DLNA

3.1.22
Device Function
A non-decomposable operational property (e.g. IP Connectivity). Device Functions should be supported by existing standards.

3.1.23
Device Option
Provides optional extensibility to existing device function requirements or is a new optional device function to the DLNA architecture.

3.1.24
Device Type
A specific device defined by its realization and its usage models. DVD players, TVs, DVRs, Mobile Phones, Printers, Cameras, Picture Frames, etc. are all examples. Device Types are used primarily for marketing descriptions, and should not be used in guideline definitions.

3.1.25
DLNA Recognized Metatadata Format
DLNA recognizes three formats of content metadata (OpenEPG, TV-Anytime, and DVB-SI).

NOTE: Mappings for elements of each of these formats are provided for certain native DLNA EPG data items.

3.1.26
DLNA Recognized Rating Authorities
DLNA recognizes certain rating authorities and their associated rating codes.

NOTE: These Rating Athorities and their codes are listed in Annex L.

3.1.27
DLNAQOS
The DLNA-defined QoS model used in this document.

3.1.28
EPG Item
a CDS item that describes a piece of content that usually does not reside on the local server. It refers to a broadcast item that is available through a tuner or it can be content accessible via a URL.

NOTE: In general there is a time associated with an EPG item that describes the broadcast time and duration of the content or the time period in which it is available for on demand retrieval via a URL. Items described by an EPG item can be recorded with the SRS service can then actually reside on a local server.

3.1.29
Exposed
Content that is listed by a UPnP AV ContentDirectory Service (CDS). Content does not necessarily exist at the time that it is exposed (for example, needs transcoding or conversion).

3.1.30
Extended Tuner
A DLNA tuner implementation based upon the TUNER Feature defined in UPnP AVv2 and higher specifications.

3.1.31
Full TS
A piece-wise constant rate MPEG-2 Transport Stream that is fully compliant with [5], 2.4.2.2. A Full TS is characterized by a consistent temporal relationship (or "even spacing") between any two adjacent TS Packets.

3.1.32
Ideal Network Conditions
The definition of a network state used only for testing and validation of guideline compliance. The effective network capacity must be substantially greater than the aggregate bandwidth of the content under test, and there are no additional devices competing for available network resources at the time of test.

3.1.33
Instance State Variable
The AVTransport and RenderingControl services define a new type of state variable, called an instance state variable. These are state variables associated with a virtual instance of a service. Both services use an evented state variable by the name of LastChange, to report a value change of an instance state variable.

3.1.34
Interactive Transfer
A media transfer mode where the target content binary is transferred at the maximum achievable rate for the communication channel and the two communicating parties. This mode is typically used for example when sending images for immediate display.

3.1.35
kHz
Refers to the number of kilohertz (1 kHz = 1,000 hertz).

3.1.36
Link-level Bearer
This definition restricts the bearer concept to include only the physical layer and link-layers as described for the ISO's OSI model.

3.1.37
Link Protection
The protection of a content stream between two devices on a DLNA network from illegitimate observation or interception.

3.1.38
Media Class
The type of media a Device Type or Device Class supports. The media classes used in this document are Image, Audio only, and Video with Audio (AV).

3.1.39
Media Collection File
A content binary whose purpose is to reference other content binaries usually for sequenced playback. Media collection files are often used for audio or AV playlists or image slideshows.

3.1.40
Media Format
The format type for content of a Media Class that is exposed by a UPnP AV MediaServer contained in a device that acts as a DMS. Examples for the Media Classes are: Image - JPEG; Audio - LPCM; AV - MPEG-2.

3.1.41
Media Stream
As used for the RTP Media Transport: A single elementary stream or MPEG-2 TS or MPEG-2 PS multiplex.

3.1.42
Media Transfer Mode
The type of transfer used to deliver content from a Content Source to a Content Receiver. There are three types of DLNA media transfer modes: Streaming Transfer, Interactive Transfer, and Background Transfer.

3.1.43
Native Device
If a device creates a virtual server and adds functionality to an existing server in the network, the existing server in the network is known as the Native Server. If a device creates a virtual

renderer and adds functionality to an existing renderer in the network, the existing renderer in the network is known as the Native Renderer. Either a Native Server or Native Renderer is known as a Native Device. These Native Devices are in contrast to a Virtual Server or Renderer (see definition below).

3.1.44
Network De-Jitter Buffer
As used for the RTP Media Transport: Buffer space that is used to store data for de-jittering and de-interleaving. This includes RTP header and payload.

3.1.45
Non-Cacheable Content
ontent binaries whose binary representations are valid only for one particular transaction at a particular instant of time are considered non-cacheable content. Intermediate HTTP caches need directives to prevent the default caching of such content. In these Interoperability Guidelines non-cacheable content includes (but it is not limited to):

- Live TV streams.
- Content resulting from transcoding or encoding operations when the output binaries have been optimized for a particular transaction (e.g. encoding to match the channel conditions for a given transaction).

3.1.46
Non-Streamable Channel Object
Non-Streamable Channel Objects are non-streamable CDS objects (i.e. no res property value) which represent a single channel of a broadcast source which presents content in a “channelized” format.

Since these CDS items are not streamable, a res property value is not needed. But having a res property without a URI value can be useful by UPnP AV MediaServer control points in determining the DLNA media format profile for the Non-Streamable Channel Object by examining the res@protocolInfo property’s DLNA.ORG_PN value in the fourth field.

3.1.47
Open-end Recording
A recording which the scheduled duration is not specified for. The duration of an open-end recording is determined by the UPnP AV MediaServer.

3.1.48
Partial SPTS
A partial MPEG-2 Transport Stream which is also a Single Program Transport Stream (SPTS).

3.1.49
Post-decoder Buffer
As used for the RTP Media Transport: The buffer space used to store decompressed data before rendering.

3.1.50
Pre-decoder Buffer
As used for the RTP Media Transport: The hypothetical reference decoder buffer that is used to contain a media (audio/video) stream after it has arrived from the network and before it is decoded into a renderable frame.

3.1.51
Primary ProtocolInfo Set
The primary protocolInfo set is defined as the first, second, and third fields of protocolInfo plus the additional DLNA.ORG_PN value which appears in the fourth field. In other words, the

primary set defines a transport method (first value), a mime type (third value), and a DLNA media format profile.

3.1.52
Printer
A Printer is a device defined in [89]. Printers are required to provide a printer service called PrintEnhanced and can optionally provide other services (e.g. a fax service). In the case of the DLNA DMPr Device Class the required print service is PrintEnhanced:1.

3.1.53
Printing Controller
A DLNA Device Capability that contains a UPnP control point that allows the user to select the content they want to print, select the Printer they want to use and monitor the status on the job. The Printing Controller will be the component that composes the XHTML-Print document that is then passed to the Printer. A Printing Controller can be the source of Image content (+PR1+), The Device Capability acronyms used in these guidelines are +PR1+ and +PR2+.

3.1.54
Receiver Buffer
As used for the RTP Media Transport: The total buffer space used to store data received from the server before the decoding.

3.1.55
Receiving Endpoint
Defined specifically through guideline 7.5.4.4.2.4, which requires an RTP client and an RTSP client.

3.1.56
Rendering Endpoint
Content Receiver devices with the capability of rendering the content they receive. These devices could play the content at the time of the transfer, right after the transfer has finished, or at a later time after the transfer has finished. For the purpose of this specification, devices in the following Device Classes constitute the only known Rendering Endpoints: DMP, DMR, DMPr, M-DMP, M-DMD.

3.1.57
Remote UI
A user interface provided by an application on a server device, that can be rendered by one or more client devices.

3.1.58
RTP Media Transport
The transport mechanism for real-time media streams between DLNA device classes and capabilities. This transport mechanism uses RTP, RTCP, RTSP, SDP protocols, and RTP payload formats with their associated media profiles.

3.1.59
RTP Session
One or more RTP Streams that are transmitted to the same destination IP address and UDP port. Typically, there is a one-to-one mapping between RTP Streams and RTP Sessions, but it is possible for multiple RTP Streams to use the same RTP Session (port multiplexing). Note that associated RTCP traffic is also part of that RTP Session although the packets are sent to the next higher UDP port number.

3.1.60
RTP Stream
A Media Stream that is encapsulated in RTP. All of the RTP packets have the same SSRC and are transmitted on the same RTP Session.

3.1.61
RTSP Session
A complete RTSP "transaction", e.g., the viewing of a movie. A session typically consists of a client creating one or more RTP Sessions (SETUP), starting the stream with PLAY or RECORD, and closing the RTSP Session with TEARDOWN. RTSP Sessions are identified using the RTSP "Session" header.

3.1.62
Scaling
Changing the visual extent of an image or video portion of an AV media stream.

3.1.63
Serving Endpoint
Defined specifically through guideline 7.5.4.4.2.3, which includes the RTP server and RTSP server.

3.1.64
Streamable Channel Object
A Non-Streamable Channel Object with one or more <res> properties containing valid URI values for an Extended Turner. This enables content for a tuner channel to be streamed over the network.

3.1.65
Streaming Transfer
For Audio and AV streams the term Streaming Transfer indicates that the packets are transferred minimally at a rate sufficient for real-time rendering. Note: This definition does not imply that the receiving endpoint will always render content exchanged using Streaming Transfer (e.g., it might store the content)

3.1.66
System Usage
Describes a device interaction model between Device Classes and/or Device Capabilities. System Usages are derived when enabling home networking use case scenarios.

3.1.67
Test Conditions
Test Conditions" are when all of the following conditions are satisfied:

- Only 1 Content Receiver and only 1 Content Source.
- Only 1 active connection at any given time.
- Both devices (Content Receiver and Content Source) must be configured by vendors before testing begins to have sufficient resources to successfully complete the stated test requirements.
- Devices interact with each other under Ideal Network Conditions.

3.1.68
Thrashing
Thrashing can occur in a synchronization system that consists of a server and multiple clients. Thrashing is a deadlocked system state where the clients are changing items on the server multiple times in the same way without user intervention. For example in a system with a server (S) containing a single item (I) and two clients (A and B), if client A makes a change to I and that change causes client B to change I, thrashing can occur if A and B now continue to make the same changes that they had previously to item I without user intervention or authorization. Because the change from A generates the automatic change from B which generates a repeat of the change from A, the system is deadlocked.

3.1.69
Time-based Seek Operations
This process identifies the Rendering Endpoint request and the Content Source response that allows the former to obtain a segment of the content for playback, where such a segment is specified in units of time.

3.1.70
Transcoding
Changing the coding system used for a content binary.

3.1.71
Transrating
Changing the rate or compression parameters used within the coding system for a content binary.

3.1.72
UPnP
Architecture for pervasive peer-to-peer network connectivity of devices of all form factors. It is designed to bring easy-to-use, flexible, standards-based connectivity to ad-hoc or unmanaged networks whether in the home, in a small business, public spaces, or attached to the Internet. It is a distributed, open networking architecture that leverages TCP/IP and Web technologies to enable seamless proximity networking in addition to control and data transfer among networked devices in the home, office, and public spaces.

3.1.73
Virtual Instance
A virtual instance is the mechanism by which a UPnP device can have multiple instances of the same UPnP service. Control points that interact with the AVTransport and RenderingControl services must interact with a virtual instance of the service. Each virtual instance is identified by a non-negative InstanceID value.

3.1.74
Virtual Renderer
A DMR that accepts content and sends it to another DMR within the network for rendering. A Virtual Renderer accepts a wider range of content types, formats, rates, or sizes than the Native Device.

3.1.75
Virtual Server
A DMS or M-DMS which exposes content existing on another DMS or M-DMS, possibly containing additional media types through content transformation.

3.1.76
Virtual Tuner Container
A CDS container for an Extended Tuner that contains only CDS items (Virtual Tuner Objects) that allow channels to be streamed using a single connection (i.e. switch the content stream from one channel to another over the same URI connection).

3.1.77
Virtual Tuner Object
A CDS item in an Extended Tuner that allow channels to be streamed using a single connection (i.e. switch the content stream from one channel to another over the same URI connection.

3.1.78
VLAN Tag
A field on a layer-2 packet header defined by 802.11Q [2].

3.2 Symbols and abbreviated terms

For the purposes of this document, the following symbols and abbreviated terms apply.

3.2.1
ΔTPCR
As used for the RTP Media Transport, this value is the difference in time, measured in 90 KHz clock units, between successive PCR values in the TS stream. It is represented mathematically as:

$$\Delta T_{PCR} = PCR_{n+1} - PCR_n$$

This value is utilized in timestamp equations in 7.5.4.4.5.4 Guidelines for encapsulation of MPEG-2 streams.

3.2.2
AC 3
Popularly known as Dolby Digital*, an audio format standard for delivering up to 5.1 audio channels developed by Dolby Laboratories.

3.2.3
ADU
"Application Data Unit" As used for the RTP Media Transport: The definition of an ADU is different for each media stream. For audio media streams, an ADU is typically an audio frame. For video media streams, an ADU is typically a "slice" (e.g. an NAL unit) or in some cases a complete video picture. Also as a special case when MPEG-2 TS encapsulation is used, each TS packet is an ADU.

3.2.4
ACK
"Acknowledge" Typically used to describe an action following a network packet being successfully received.

3.2.5
AP
"Access Point" A specially configured Network Infrastructure Device on a wireless local area network (WLAN). Access points act as a central transmitter and receiver of WLAN radio signals. APs used in home networks are generally small, dedicated hardware devices featuring a built-in network adapter, antenna, and radio transmitter. These APs support Wi-Fi wireless communication standards.

3.2.6
ARP
"Address Resolution Protocol" A protocol in the TCP/IP family that resolves an IP address to a hardware address, such as an Ethernet address.

3.2.7
ATSC
"Advanced Television Systems Committee" One of the standard bodies for digital television broadcasting.

3.2.8
AV
"Audio with Video" Refers to any media content that contains both moving picture and sound.

3.2.9
AVP
"Audio/Visual Profile" As used for the RTP Media Transport.

3.2.10
AVPF
"Extended Audio/Visual Profile for RTCP-based Feedback" As used for the RTP Media Transport.

3.2.11
AVT
"AVTransport Service" The AVTransport Service is a UPnP service that provides network-based control for common transport operations such as play, stop, pause, next, previous, and seek. The AVTransport Service specification is a standard UPnP DCP.

3.2.12
BD_ADDR
"Bluetooth Device Address" A unique 48-bit Bluetooth Hardware address.

3.2.13
BNEP
"Bluetooth Network Encapsulation Protocol" A packet format for Bluetooth network encapsulation used to transport common networking protocols over the Bluetooth media. Bluetooth network encapsulation supports the same networking protocols that are supported by IEEE 802.3/Ethernet encapsulation.

3.2.14
BT
"Bluetooth"

3.2.15
BK
"Background User Priority" A priority-based QoS level for Background User Priority.

3.2.16
BE
"Best-Effort User Priority" A priority-based QoS level for Best-Effort User Priority.

3.2.17
CDB
"Coded Data Buffer" As used for the RTP Media Transport. Buffer space that is used to store compressed data before decoding.

3.2.18
CDS
"ContentDirectory Service" The ContentDirectory Service is a UPnP service that provides network-based discovery of content. The ContentDirectory Service specification is a standard UPnP DCP.

3.2.19
CE
"Consumer Electronics" A class of devices used in the home, such as DVD, DVR, PVR, PDA, TV, set top box, cellular phones, …

3.2.20
CMS
"ConnectionManager Service" The ConnectionManager Service is a UPnP service that provides information about the supported transport protocols and media formats of a UPnP device. The ConnectionManager Service specification is a standard UPnP DCP.

3.2.21
CNAME
"Canonical Name" As used for the RTP Media Transport.

3.2.22
CP
"UPnP Control Point" Generic reference to any UPnP control point.

3.2.23
CSRC
"Contributing Source" As used for the RTP Media Transport.

3.2.24
CSS
"Cascading Style Sheets" Format defined by W3C to add style information to documents.

3.2.25
DA
"Device Architecture 1.0" UPnP Device Architecture version 1.0 document.

3.2.26
DCP
"Device Control Protocol" A specification that is standardized by the UPnP Forum. Related specifications produced by a UPnP working committee are often identified by the working committee name. For example, UPnP AV 1 DCP.

3.2.27
DDC
"Device Discovery and Control" A subclause heading in the Interoperability Guidelines that defines the underlying interoperability architecture for the discovery and control devices.

3.2.28
DH1-5
"Data - High Rates 1 through 5" Bluetooth specific physical layer packet types.

3.2.29
DHCP
"Dynamic Host Configuration Protocol" A protocol to automatically provide IP addresses and other network configuration information to network nodes.

3.2.30
DIDL
"Digital Item Declaration Language" An XML schema for representing the metadata of digital content.

3.2.31
DIDL-Lite
"Digital Item Declaration Language - Lite" An XML schema used by the UPnP Forum for representing the metadata of digital content. The XML schema uses a subset of the DIDL schema with additional metadata properties defined by the UPnP Forum.

3.2.32
DLNA
"Digital Living Network Alliance" The organization that created this document.

3.2.33
DLNAQOS_UP
"DLNA QoS User Priority" A DLNA-defined QoS label used to correlate an underlying 802.1Q User Priority and WMM Access Category to a DLNA Traffic Type(s).

3.2.34
DM1-5
"Data - Medium Rates 1 through 5" Bluetooth specific physical layer packet types.

3.2.35
DMC
"Digital Media Controller" A DLNA Device Class having home network environmental characteristics, with the role of finding content exposed by a DMS and matching it to the rendering capacities of a DMR and setting up the connections between the DMS and the DMR.

3.2.36
DMP
"Digital Media Player" A DLNA Device Class having home network environmental characteristics, with the role of finding content exposed by a DMS and rendering the content locally.

3.2.37
DMPr
"Digital Media Printer" A DLNA Device Class having home network environmental characteristics, with the role of providing document and image printing services to other DLNA devices.

3.2.38
DMR
"Digital Media Renderer" A DLNA Device Class having home network environmental characteristics, with the role of rendering content it receives after being setup by another network entity.

3.2.39
DMS
"Digital Media Server" A DLNA Device Class having home network environmental characteristics, with the role of exposing and distributing content throughout the home.

3.2.40
DOM
"Document Object Model" Interface defined by W3C to dynamically access and update the content, structure and style of XML documents, including XHTML documents.

3.2.41
DNS
"Domain Name System" A protocol that enables hierarchical names for Internet domains and addresses. The protocol includes the means to translate between numerical IP addresses and text host names.

3.2.42
DSCP
"Differentiated Services (DiffServ) Code Point" A QoS field, defined by the DiffServ discipline [27], found in the layer 3 header of IP packets.

3.2.43
DVB
"Digital Video Broadcasting"   One of the standard bodies for digital television broadcasting.

3.2.44
DVD
"Digital Versatile Disc"   A high capacity multimedia data storage medium.

3.2.45
DVR
"Digital Video Recorder"   A consumer electronic device.

3.2.46
ES
"Elementary Stream" A general term for a coded video, coded audio or other coded bitstream.

3.2.47
HD
"High Definition"   Picture quality at an HDTV level.

3.2.48
HDTV
"High Definition Television"   Provides a higher quality display, with a vertical resolution display from 720p to 1080i and higher and an aspect ratio (the width to height ratio of the screen) of 16:9, for a viewing experience similar to watching a movie.

3.2.49
HID
"Home Infrastructure Device"   A Device Category that groups together all the applicable DLNA Device Classes, which provide support for interoperability between Device Classes of different Device Categories . Device Classes in this Device Category are: M-NCF and MIU.

3.2.50
HND
"Home Network Device"   A Device Category that groups together all the applicable DLNA Device Classes with home network environmental characteristics (requirements). Device Classes in this Device Category are: DMS, DMP, DMR, DMC, and DMPr.

3.2.51
HTTP
"Hyper Text Transfer Protocol"   A protocol for transferring files across the Internet. Requires an HTTP client program on one end, and an HTTP server program on the other end.

3.2.52
ICMP
"Internet Control Message Protocol"   A protocol in the TCP/IP family that is used for out-of-band messages related to network operation.

3.2.53
IGD
"Internet Gateway Device"   A multifunction Network Infrastructure Device that routes and/or bridges global internet with the local area network.

3.2.54
IPR
"Intellectual Property Rights"

3.2.55
IP
"Internet Protocol"

3.2.56
IPv4
"Internet Protocol version 4" An OSI network layer 3 protocol.

3.2.57
JPEG
"Joint Photographic Experts Group" A coding standard for compression of still images (pictures).

3.2.58
L2CAP
"Logical Link Control and Adaptation Protocol" A Bluetooth Link Layer protocol.

3.2.59
LAN
"Local Area Network" Closely administered network segment(s) such as within the home or office.

3.2.60
LFE
"Low Frequency Enhancement" DVB-specified way to transmit additional sound information.

3.2.61
LMP
"Link Management Protocol" Bluetooth Link management protocol for managing communication link of Bluetooth piconet.

3.2.62
LPCM
"Linear Pulse Code Modulation" An uncompressed audio encoding.

3.2.63
M-DMC
"Mobile Digital Media Controller" A DLNA Device Class having mobile handheld environmental characteristics, with the role of finding content exposed by an M-DMS and matching it to the rendering capacities of a DMR and setting up the connections between the M-DMS and the DMR.

3.2.64
M-DMD
"Mobile Digital Media Downloader" A DLNA Device Class having mobile handheld environmental characteristics, with the role of downloading content from an M-DMS.

3.2.65
M-DMP
"Mobile Digital Media Player" A DLNA Device Class having mobile handheld environmental characteristics, with the role of finding content exposed by an M-DMS and rendering the content locally.

3.2.66
M-DMS
"Mobile Digital Media Server" A DLNA Device Class having mobile handheld environmental characteristics, with the role of exposing and distributing content throughout the home

3.2.67
M-DMU
"Mobile Digital Media Uploader" A DLNA Device Class having mobile handheld environmental characteristics, with the role of uploading content to an M-DMS.

3.2.68
M-NCF
"Mobile Network Connectivity Function" A DLNA Device Class that provides interoperability by bridging the network connectivity layer between devices in the HND and MHD Device Categories.

3.2.69
MF
"Media Formats" This term is used to describe guidelines related to the collection of Media Format Profiles defined in [56].

3.2.70
MHD
"Mobile Handheld Device" A Device Category that groups together all the applicable DLNA Device Classes with mobile handheld environmental characteristics (requirements). Device Classes in this Device Category are: M-DMS, M-DMP, M-DMD, M-DMU, and M-DMC.

3.2.71
MHP
"Multimedia Home Platform" An optional application interface used together with MPEG-2 transmissions.

3.2.72
MIME
"Multipurpose Internet Mail Extension" A standard system for identifying the type of data contained in a file. MIME is an Internet protocol that allows sending binary files across the Internet as attachments to e mail messages. This includes graphics, photos, sound, video files, and formatted text documents.

3.2.73
MIU
"Media Interoperability Unit" A DLNA Device Class that provides media format interoperability between devices in the HND and MHD Device Categories.

3.2.74
MM
"Media Management" A subclause heading in the Interoperability Guidelines.

3.2.75
MPEG
"Moving Picture Experts Group" The name of an organization for developing standards related to audiovisual information.

3.2.76
MRCP
"MediaRenderer Control Point" A UPnP control point that issues actions to an MRD.

3.2.77
MRD
"MediaRenderer Device" The MediaRenderer Device (a.k.a. MediaRenderer) is a UPnP device that provides network-based control for the rendering of content. Minimally, a

MediaRenderer must have a RenderingControl Service and a ConnectionManager service. The MediaRenderer specification is a standard UPnP DCP.

3.2.78
MSCP
"MediaServer Control Point"   A UPnP AV control point that issues actions to an MSD.

3.2.79
MSD
"MediaServer Device"   The MediaServer Device (a.k.a. MediaServer) is a UPnP device that provides network-based discovery of content. Minimally, a MediaServer must have a ConnectionManager Service and a ContentDirectory Service. The MediaServer specification is a standard UPnP DCP.

3.2.80
MT
"Media Transport"   A subclause heading in the Interoperability Guidelines.

3.2.81
NAL Unit
"Network Abstraction Layer Unit"   Term is specific to H.264 RTP payload format. It is a 1 byte header and the payload byte string (as defined in [43]).

3.2.82
NAP
"Network Access Point" When using the Bluetooth PAN profile (with BNEP protocol), a NAP acts as an Ethernet bridge relaying link layer traffic (Ethernet packets) between PAN devices and another network.

3.2.83
NC
"Networking and Connectivity"   A subclause heading in the Interoperability Guidelines.

3.2.84
NC-PS
"Network Connectivity Power Saving" Specifies abstract power states for a communication link in power management actions.

3.2.85
NID
"Network Infrastructure Device"   Devices which provide supporting functionality in the home network such as access points, bridges, Internet gateways, routers, and switches, but which are not members of the normative HID Device Category. These devices facilitate a good user experience with DLNA devices but are only covered at this time in this document by informative recommendations in Annex A.

3.2.86
NTSC*
"National Television Systems Committee"   A standard for broadcast and reception of analog television signals.

3.2.87
OSI
"Open Systems Interconnection"   Networking stack model (7 layers).

3.2.88
PAL*
"Phase Alternating Line"   A standard for broadcast and reception of analog television signals.

3.2.89
PAN
"Personal Area Network" The Bluetooth profile that allows IP-based communication over a Bluetooth network.

3.2.90
PANU
"PAN User" A Bluetooth device that participates in a Bluetooth PAN network set up by a NAP.

3.2.91
PC
"Personal Computer" A general-purpose computer equipped with a microprocessor and designed to run commercial software (such as a word processor or World Wide Web browser) for an individual user.

3.2.92
PCR
"Program Clock Reference" Refer to MPEG-2 standard 13818-1[5].

3.2.93
PNG
"Portable Network Graphics" It is a coding standard for compression of still images (pictures).

3.2.94
PrCP
"Printer Control Point" UPnP control point that issues actions to a PrD.

3.2.95
PrD
"Printer Device" The Image Printer Device is a UPnP device that provides network-based control for the printing. Minimally, a Printer Device must have a PrintEnhanced Service. The Printer Device specification is a standard UPnP DCP

3.2.96
PS
"Program Stream" Usually in reference to an MPEG-2 AV stream format.

3.2.97
PVR
"Personal Video Recorder" A consumer electronic device.

3.2.98
QoS
"Quality of Service" To provide guarantees on the ability of a network to deliver predictable results.

3.2.99
RCS
"RenderingControl Service" The RenderingControl Service is a UPnP service that provides network-based control for the adjustment of rendering attributes such as volume, brightness, contrast, and mute. The RenderingControl Service specification is a standard UPnP DCP.

3.2.100
RTP
"Real Time Protocol" Media transport that provides end-to-end network transport functions for transmitting real-time data, such as AV. It provides services such as payload type identification, sequence numbering, time-stamping, and delivery monitoring.

3.2.101
RTCP
"Real Time Control Protocol"

3.2.102
RTSP
"Real Time Streaming Protocol"

3.2.103
RUI
"Remote User Interface" A user interface provided by an application on a server device, that can be rendered by one or more client devices.

3.2.104
SCPD
"Service Control Protocol Description" This is an XML-encoded file describing a UPnP service. This is also known as a service description file.

3.2.105
SDES
"Session Description Item"

3.2.106
SDP
"Session Description Protocol"

3.2.107
SOAP
"Simple Object Access Protocol" An XML based messaging protocol used to exchange service requests and responses over a network.

3.2.108
SPTS
"Single Program Transport Stream"

3.2.109
SSDP
"Simple Service Discovery Protocol" UPnP device discovery protocol.

3.2.110
SSRC
"Synchronization Source"

3.2.111
TCP
"Transmission Control Protocol" A protocol in the TCP/IP family used for the reliable exchange of data over a network.

3.2.112
TIFF
"Tagged Image File Format" Media format of still images (pictures).

3.2.113
TS
"Transport Stream" Usually in reference to an MPEG-2 AV stream format.

3.2.114
TTS
"Timestamped Transport Stream" A Transport Stream which is composed of timestamped TS packet. The timestamped TS packet is a 192-byte packet consisting of a 188-byte ISO MPEG-2 TS packet plus a 4-byte timestamp in advanced of the TS packet.

3.2.115
UDP
"User Datagram Protocol" A protocol in the TCP/IP family used for the unreliable exchange of data over a network.

3.2.116
UP
"User Priority" A 3-bit field of the QoS Control field in the WMM Specification [85] and a Tag control information field in the VLAN tag header of 802.1Q [2] that defines the relative priority of a packet.

3.2.117
URI
"Uniform Resource Identifier" The W3C's codification of the name and address syntax of present and future objects on the Internet. In its most basic form, a URI consists of a scheme name (such as file, http, ftp, news, mailto, gopher) followed by a colon, followed by a path whose nature is determined by the scheme that precedes it. URI is the umbrella term for URNs, URLs, and all other Uniform Resource Identifiers.

3.2.118
URL
"Uniform Resource Locator" A type of URI.

3.2.119
UTF
"Unicode Transformation Format"

3.2.120
VI
"Video User Priority" A priority-based QoS level for Video User Priority.

3.2.121
VLAN
"Virtual Local Area Network"

3.2.122
VO
"Voice User Priority" A priority-based QoS level for Voice User Priority.

3.2.123
WAN
"Wide Area Network" Usually in reference to the network outside the home or office. Typically in reference to the entire global Internet.

3.2.124
WMM
"Wi-Fi Multimedia" WMM is the name used to describe the QoS Guidelines and associated Test Plans published by the Wi-Fi Alliance for 802.11 wireless networks.

3.2.125
XML
"Extensible Markup Language" A text-based declarative language used to describe structured data for information exchange.

4 DLNA Home Network Architecture

To achieve interoperability between connected digital media devices in the home, a common set of building blocks based on existing standards is needed as a basis to develop the DLNA Home Networked Device Interoperability Guidelines. Table 1 shows the specific functional components and technology ingredients that are covered in the Interoperability Guidelines. Figure 1 illustrates these functional components within the networking architecture of the Interoperability Guidelines. The Interoperability Guidelines define usage of these functional components to ensure interoperability among Device Classes defined in Clause 5. A brief overview of each functional component follows in the subsequent subclauses.

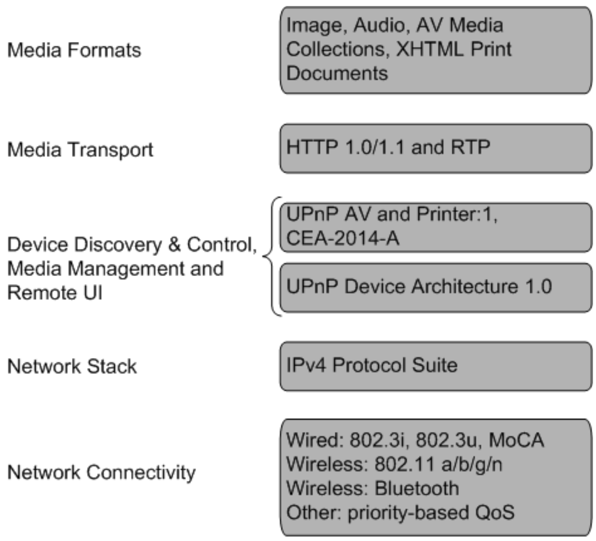


Figure 1 — DLNA Functional Components

4.1 Networking and Connectivity

The IPv4 protocol suite is the foundation for networking and connectivity for DLNA devices in the digital home. IP also provides the underlying network communications for applications on the Internet. Based on industry-standard specifications from the IETF, IP is implemented and supported in a wide range of devices. IP has several advantages for use by DLNA devices:

- IP has demonstrated that it allows applications to run over different network topologies transparently.
- IP allows connecting every device in the home to the Internet.
- IP connectivity solutions are widely used and are cost effective. The most common ones are Ethernet (802.3i and 802.3u) and wireless technologies (802.11a, 802.11b, and

802.11g) for devices in the home networking environment. In the mobile handheld environment, Bluetooth is the prevalent wireless technology in use.

Subclause 7.2 specifies the detailed guidelines to enable interoperability between DLNA devices in the digital home. In addition, the home environment requires supporting network infrastructure, such as access points, bridges, Internet gateways, routers, and switches. These non-normative devices are referred to in this document as Network Infrastructure Devices (NID). Annex A provides informative recommendations for Network Infrastructure Devices to facilitate a good user experience and interoperability with DLNA devices.

4.1.1 Network Quality of Service

Multimedia applications on IP networks benefit from Quality of Service (QoS) functionality to optimize the way shared network resources are allocated among different applications. Without QoS, all applications running on different devices have an equal opportunity to transmit data frames. Multimedia applications such as video streaming and music streaming are sensitive to excessive latency variations and throughput reductions. With prioritized QoS, applications label (tag) packets to indicate the User Priority (UP) that dictates how the packets are allowed to access network resources.

The DLNA QoS model is intended to allow DLNA applications that wish to take advantage of User Priority to have common usage rules for tagging. Devices that do not wish to use QoS shall be tolerant of tagging. The DLNA QoS model promotes fair and consistent usage of priorities and balanced performance across all DLNA Traffic Types, in addition to interoperability; thus enhancing the overall user experience.

4.2 Device Discovery and Control

Device discovery and control enables a device on the home network to discover the presence and capabilities of other devices on the network and collaborate with these devices in a uniform and consistent manner. The UPnP Device Architecture, version 1.0, addresses all of these needs and simplifies device networking in the home. For this reason, UPnP Device Architecture is the device discovery and control solution for DLNA devices. Subclause 7.2.5.5.27.6 specifies the detailed guidelines to enable interoperability between DLNA devices in the digital home.

4.3 Media Management

Media management enables devices and applications to identify, manage, and distribute media content across the home network devices. UPnP Audio/Video (AV) and UPnP Printer technology addresses all of these needs for the home network and is the media management solution for DLNA devices.

The UPnP AV architecture defines the interaction model between UPnP AV devices and associated control point applications. UPnP AV devices can instantiate themselves in a variety of form factors, including (but not limited to) TVs, VCRs, DVD players, Set-Top Boxes, stereo systems, still-image cameras, portable media players, cell phones, and PCs. The UPnP AV architecture allows devices to support entertainment content in any format using any media transfer protocol. The UPnP AV specification defines two types of UPnP devices on the home network: UPnP AV MediaServers and UPnP AV MediaRenderers. The specifications also define four services hosted by UPnP AV MediaServers and UPnP AV MediaRenderers. The existence of UPnP control points that interact with UPnP AV devices and services is implied.

1. Content Directory Service: Exposes the available content.
2. Connection Manager Service: Determines how the content can be transferred from the UPnP AV MediaServer to the UPnP AV MediaRenderer devices.
3. AV Transport Service: Controls the flow of the content.

4. Rendering Control Service: Controls how the content is played.

The UPnP Printer architecture defines the interaction model between UPnP printing devices and associated control point applications. Examples of UPnP Printer devices are photo printers. The UPnP Printer architecture defines a UPnP Printer device which specifies the UPnP PrintEnhanced:1 service which in turn specifies XHTML-Print, CSS Print, and CSS Print enhanced layout extension as the page description language.

See clause 5 for further information on how UPnP technology components are mapped into DLNA Device Classes.

Subclause 7.4 specifies the detailed guidelines to enable interoperability between DLNA devices in the digital home.

4.4 Media Formats

Media formats describe how content is encoded and formatted for transport and rendering on the home network. The DLNA media format model is intended to achieve a baseline for network interoperability while encouraging continued innovation in media codec technology. For each Device Category, the DLNA media format model defines a set of mandatory and optional media format profiles for each of the three classes of media: imaging, audio, and AV. A media format profile is a set of attributes, parameters, and system and compression level details sufficient to describe the media format of a content binary to enable interoperability between DLNA devices in each Device Category. In order to support interoperability between devices of different Device Categories, the Media Interoperability Unit performs basic translation between the required media format profiles of different Device Categories. In addition, the DLNA media format model specifies rules about conversion between optional and mandatory formats to ensure that content can be enjoyed on all devices. The Interoperability Guidelines Media Formats part [56] specify the detailed guidelines to enable interoperability between DLNA devices in the digital home.

4.5 Media Transport

Media transport defines how content travels across the home network. DLNA devices that source or receive media content across the home network shall support HTTP as the baseline transport mechanism for the transfer of content. In addition, the RTP transport can optionally be used as a media transport; but the mandatory requirements for HTTP shall always be supported. Subclause 7.5 specifies the detailed guidelines to enable interoperability between DLNA devices in the digital home.

4.6 Remote UI

Remote UI defines how UI content is described, formatted, and transported from one device to another over the network. This also includes mechanisms for sending events and UI updates between different devices. DLNA adopts CEA-2014-A [101] as the baseline technology for achieving this. Device discovery and control in CEA-2014-A is based on the UPnP Device Architecture version 1.0, and hence reuses the DLNA Device Discovery and control guidelines, the specifics of which can be found in 7.3. Transport of UI content is achieved by using an extended subset of XHTML (called CE-HTML) over HTTP. More details about this can be found in 7.3. This also includes guidelines as to how to integrate remote UI with A/V Media.

5 DLNA Device Model

5.1 Overview

These guidelines address the requirements of devices with differing environmental characteristics, such as home network and mobile handheld devices. Home Network Devices (HNDs) and Mobile Handheld Devices (MHDs) are Device Categories that have a differing set of requirements in media formats and network connectivity. This clause provides a device model with consistent terms and usages for these Device Categories. To support interoperability between Home Network Devices and Mobile Handheld Devices, it is possible for a Home Network Device to meet all the requirements for the corresponding Mobile Handheld Device. It is also possible for a Mobile Handheld Device to meet all the requirements for the corresponding Home Network Device. In these cases, such a device is a member of both the HND and a MHD Device Categories. But in most cases that might not be feasible, so another way to achieve interoperability is via a group of devices that will be able to provide bridging or content transformation services between these two Device Categories. These devices belong to a Device Category referred to as the Home Infrastructure Device (HID). The following summarizes these devices:

- Bridging network connectivity between the MHD and the HND Device Categories (Figure 24).
- Media format interoperability services between the MHD and the HND Device Categories (Figure 33).

In summary, the key points about Device Categories are:

- Each is uniquely optimized for the requirements of a particular environment.
- The device guidelines focus on interoperability of devices within a Device Category.
- There are guidelines for devices which facilitate interoperability between Device Categories.
- A device can choose to be a member of multiple Device Categories.

5.2 Device Model Elements

As described in clause 4, devices adhering to the DLNA Home Networked Device Interoperability Guidelines have six architectural layers. In summary, they are Media Formats for describing conformant content, Media Management for describing how content is found and controlled to achieve different System Usages, Device Discovery and Control for device control, Media Transport for the transfer of content, Network Stack for IPv4 protocol requirements, and Network Connectivity for supporting different network physical layers. The following are terms used within this clause and throughout the guidelines. Their interdependence is illustrated in Figure 2.

A Device Category is an aggregation of Device Classes with common environmental characteristics (e.g. mobile devices) and sharing System Usages that enable home networking use case scenarios. An example of a Device Category is the set of all Device Classes with System Usages that solve requirements in media formats and network connectivity in a home network environment, such as a HND (Home Networked Device). Device Classes are grouped within a Device Category, but a single physical device can fall into multiple Device Categories.

A System Usage describes a device interaction model between Device Classes and/or Device Capabilities. System Usages are derived when enabling home networking use case scenarios. An example is a rendering device with a dedicated user interface that is browsing, selecting, and playing content from a media server on the home network, such as the 2-Box Pull System Usage

A Device Class is a set of Device Functions (at least one) aggregated to be used in a System Usage that enables home networking AV use case scenarios. A Device Class shall provide support for all layers in the DLNA architecture except when defined within the Home Infrastructure Device Category which is providing interoperability between one or more layers in the DLNA architecture. A Device Class is a certifiable entity by DLNA and is derived from System Usages. It specifies the capabilities supported by a device regardless of the device's physical attributes. An example of a Device Class is a device with the role of exposing and distributing content throughout the home such as a "DMS". A single physical device can support multiple Device Classes.

A Device Capability is a set of Device Functions (at least one) aggregated to be used in a System Usage that is enabling home networking AV use case scenarios. A Device Capability does not provide support for all layers in the DLNA architecture. It typically contains Device Functions at the Device Discovery and Control, Media Management, and Media Transport layers only. A Device Capability is not a Device Class and cannot stand alone. It shall always be deployed in conjunction with an implementation of a valid Device Class. A Device Class might already contain some of the Device Functions required to provide a Device Capability. An example of a Device Capability is any DLNA device that incorporates the additional feature (capability) of pushing content to a rendering device, such as a "Push Controller".

A Device Function is a non-decomposable operational property. Device Functions should be supported by existing standards or specifications. A Device Function usually applies to a single layer within the DLNA architecture. An example of a Device Function is an operational component at the Device Discovery and Control layer of the DLNA architecture such as a "UPnP Device". Device Classes and Device Capabilities are composed of a set of Device Functions. Device Functions are the building blocks of DLNA devices.

A Device Option provides optional extensibility to an existing Device Class definition, such as upload or scheduled recording functionality added to a MediaServer Device (MSD), or it provides a new optional Device Function to the DLNA architecture, such as RTP. A Device Option differs from a Device Class or Device Capability in that it normally enables a Device Class or Device Capability to perform an existing System Usage in a different way (e.g., RTP). In the case where a Device Option achieves a new System Usage, it adds functionality to an existing Device Class or Device Capability to support a new interaction such as the Upload System Usage.

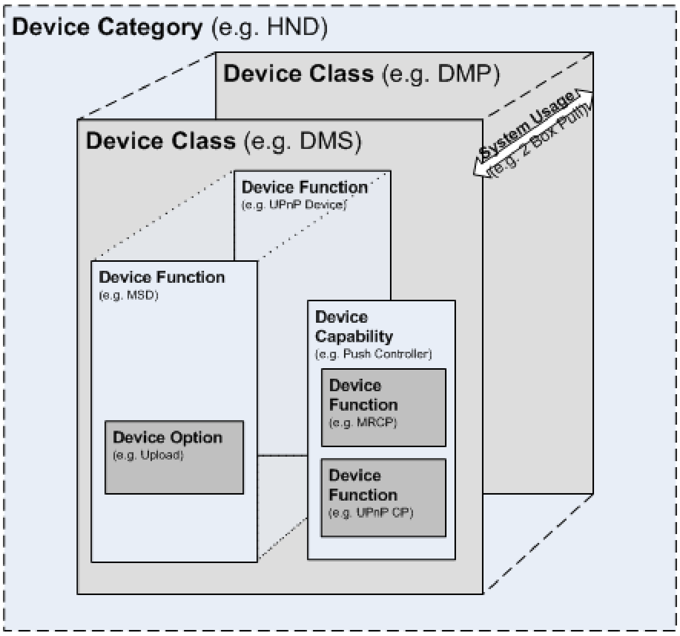


Figure 2 — DLNA Device Model Terms Hierarchy

5.3 Device Functions

For the Interoperability Guidelines and System Usages, the Device Functions below are defined for the DLNA architecture.

- IP Connectivity - Network Connectivity and Network stack (7.2). This incorporates Ethernet (802.3), 802.11, and/or Bluetooth connectivity and IP networking using the IPv4 protocol suite.
- UPnP Device and UPnP Control Point (UPnP CP) - Device Discovery and Control based upon the UPnP Device Architecture (7.2.5.5.27.6). This incorporates the baseline device architecture used by all Device Classes and Device Capabilities.
- UPnP AV MediaServers (MSD), UPnP AV MediaServer control point (MSCP), UPnP AV MediaRenderer (MRD), UPnP AV MediaRenderer control point (MRCP), UPnP Printer Device (PrD), and UPnP Printer control point (PrCP) - Media Management (7.4). This incorporates the control functionality that is layered on top of a UPnP Device or a UPnP control point to fulfill a role for a Device Class or a Device Capability in a System Usage. An MSD provides methods to access to content. An MSCP is a controller that can browse and select content provided by an MSD. An MRD provide methods to render content. An MRCP is a controller that selects the content to be rendered by an MRD. A PrD provides the ability to print Image content. A PrCP is a controller that creates print jobs for selected content to be printed by a PrD.
- RUI Server (RUIS), RUI Client (RUIC), RUI Server Control Point (RUIS-CP), RUI Client Control Point (RUIC-CP), UI Transport Server (UITS), UI Transport Client (UITC), UI Content and RUI Presenter - Remote User Interfaces (7.8). This incorporates the functionality for control, transport and rendering of remote user interfaces in the network, as required by the remote UI related System Usages defined in 5.7. An RUIS provides UPnP device functionality to offer one or more remote user interfaces. An RUIS-CP is a controller for browsing and selecting a remote UI offered by a RUI server. An RUIC provides UPnP device functionality for exposing RUI capabilities and handling RUI actions. An RUIC-CP is a controller for setting up the connection between a RUI Client and a remote UI offered by a RUI Server. A UI Transport Server and a UI Transport Client are the device functions for transport of the UI content between a client and server. An RUI Presenter provides functionality on a RUI client to render and interact with the remote UI content coming from the RUI server.
- Media Transport Server and Media Transport Client - Media Transport (7.5). These are the Device Functions for the transport of content. The mandatory transport for content is HTTP which has the components of an HTTP Server and an HTTP Client. An optional transport for content is RTP which has the components of an RTP Serving Endpoint and an RTP Receiving Endpoint. RTP is an example of a Device Option which provides optional extensibility to System Usages utilizing a Media Transport.
- Content - Interoperability Guidelines Media Formats part [56]. This document defines the DLNA mandatory and optional Media Format Profiles for content.

5.4 Device Categories

Device categories are a grouping of Device Classes that share common environmental characteristics (requirements) with System Usages. There were no Device Categories explicitly defined in version 1.0 of the Interoperability Guidelines, as all of the Device Classes operated in the same environment. All of the version 1.0 guidelines were defined as if the following Device Category applied:

- Home Network Devices (HNDs) are a group of Device Classes that share System Usages in the home network with the same media format and network connectivity requirements.

In these Interoperability Guidelines, the following two additional Device Categories are defined:

- Mobile Handheld Devices (MHDs) are a group of Device Classes that share the same System Usages as the HND Device Category, but have different requirements for media format and network connectivity.
- Home Infrastructure Device (HID) supports interoperability between Device Categories.

5.5 Device Classes and Roles

In version 1.0 of the Interoperability Guidelines, the following two Device Classes were defined to support the 2-Box Pull System Usagefor the HND Device Category.

- A Digital Media Server (DMS) with the role of exposing and distributing content.
- A Digital Media Player (DMP) with the role of finding content exposed by a DMS and playing the content locally on the DMP.

In these Interoperability Guidelines, the following three additional Device Classes are defined for the HND Device Category.

- A Digital Media Renderer (DMR) with the role of playing content it receives after being setup by another network entity.
- A Digital Media Controller (DMC) with the role of finding content exposed by a DMS and matching it to the rendering capacities of a DMR and setting up the connections between the DMS and DMR.
- A Digital Media Printer (DMPr) with the role of printing images.

The following Device Classes are defined for MHD Device Category.

- A Mobile Digital Media Server (M-DMS) with the role of exposing and distributing content.
- A Mobile Digital Media Player (M-DMP) with the role of finding content exposed by an M-DMS and playing the content locally on the M-DMP.
- A Mobile Digital Media Uploader (M-DMU) with the role of sending content to an M-DMS with upload functionality.
- A Mobile Digital Media Downloader (M-DMD) with the role of finding and downloading content exposed by an M-DMS and playing the content locally on the M-DMD after downloading.
- A Mobile Digital Media Controller (M-DMC) with the role of finding content exposed by an M-DMS and matching it to the rendering capabilities of a DMR and setting up the connections between the server and renderer.

Many of these mobile Device Classes have counterparts in the HND Device Category; however, they differ from their counterpart at the network connectivity layer and at the media format layer in the DLNA architecture. For example, an M-DMC can be connected via mobile specific network connectivity while a DMC has to meet the HND network connectivity requirements. This should not be taken to imply that MHD and HND devices cannot interact directly. The discussion above and in the definition of terms of the MHD and HND Device Classes, mobile Device Classes interact with other mobile Device Classes, such as an M-DMP interacting with an M-DMS. However, if the mobile and home devices have compatible network connectivity, and can exchange compatible media format profiles, nothing in these statements should be taken to imply that an M-DMP cannot directly connect to a DMS to complete a system usage. See Table 4, Table 3, and Table 6 for a listing of the required device interoperations and those that are only possible given compatible network and media format profile capabilities.

In order to allow HND and MHD devices to interact even with different network and media format profile capabilities, these guidelines define infrastructure devices of the HID device category that support interconnecting HND and MHD devices. The following Device Classes are defined within the HID Device Category.

- A Mobile Network Connectivity Function (M NCF) with the role of providing a bridging function between the MHD network connectivity and the HND network connectivity. An M NCF also provides support for security and power saving modes. If a device in the MHD Device Category has only mobile specific network connectivity, it connects to home network via an M NCF. A device in the MHD Device Category with HND compatible network connectivity can connect to the home network directly.
- A Media Interoperability Unit (MIU) with the role of providing content transformation between required media formats for the HND Device Category and the MHD Device Category. If a connected device in the MHD Device Category and a device in the HND Device Category have compatible media capabilities, they can interoperate directly without the need of an MIU. If they have different media capabilities, the MIU can be used to provide media interoperability.

The Device Functions that are incorporated in these Device Classes are illustrated in the figures that provide the details for System Usages and their respective device interaction models in 5.7.1 through 5.7.12.

5.6 Device Capabilities and Roles

In these Interoperability Guidelines, the following Device Capabilities are defined.

- A Push Controller (+PU+) with the role of pushing its local content to a DMR.
- A Printing Controller-1 (+PR1+) with the role of controlling a DMPr to print image content. A +PR1+ is also responsible for serving the image content and the XHTML-Print document to the DMPr.
- A Printing Controller-2 (+PR2+) with the role of controlling a DMPr to print image content. Although a +PR2+ is not responsible for serving the image content, it is responsible for finding the image content on a DMS or M-DMS, providing the XHTML-Print document to the DMPr, and instructing the DMPr to print the images from the DMS or M-DMS.
- An Upload Controller (+UP+) with the role of sending content to a DMS or M-DMS with upload functionality.
- A Download Controller (+DN+) with the role of downloading content from a DMS or M-DMS to itself.
- An Upload Synchronization Controller (+UPSYNC+) with the role of keeping locally changing content synchronized with a DMS or M-DMS supporting the Content Synchronization Device Option.
- A Download Synchronization Controller (+DNSYNC+) with the role of keeping remotely changing content synchronized with the local system. The remotely changing content shall be located on a DMS or M-DMS supporting the Content Synchronization Device Option.
- A RUI Pull Controller (+RUIPL+) with the role of finding and loading remote UI content exposed by a +RUISRC+ capability and rendering and interacting with the UI content.
- A RUI Source capability (+RUISRC+) with the role of exposing and sourcing UI content.
- A RUI Sink capability (+RUISINK+) with the role of exposing remote UI functionality and rendering UI content it receives from a +RUISRC+ capability.
- A RUI Controller (+RUICTRL+) with the role of finding +RUISRC+ and +RUISINK+ capabilities, finding matching UIs, and setting up the connection between the +RUISINK+ and +RUISRC+.
- A Scheduled Recording Controller (+SR+) with the role of instructiong a DMS or M-DMS to browse, create, modify, and/or cancel scheduled recordings of content.
- An EPG Controller (+EPG+) with the role of fetching EPG metadata from a DMS or M-DMS.

The Device Functions that are incorporated in these Device Capabilities are illustrated in the figures that provide the details for System Usages and their respective device interaction models in 5.7.1 through 5.7.12.

5.7 System Usages

In describing the flow of content in the System Usages, indicated by the largest arrow in the System Usage diagrams in 5.7.1 through 5.7.12, the terms push and pull are used. The terms push and pull are used in System Usages to characterize the user's perception of the source or sink location in the process of content transfer. That is, pull means that the content is traveling to the user, while push means that the content is traveling from the user. This perception is not a reflection of the technical underling transport mechanism utilized to perform the transfer the content from the source to the sink (e.g. HTTP and RTP).

In these Interoperability Guidelines, the following seven System Usages are defined that map to all of the use case scenarios being enabled by the detailed guidelines.

- 2-Box Pull System Usage

  This usage involves a user at a DMP or an M-DMP, which enables the user to find and play content that is advertised and distributed by a DMS or M-DMS.

- 2-Box Push System Usage

  This usage involves a user at a Push Controller, which enables the user to distribute content to a DMR for playback purposes.

- 3-Box System Usage

  This usage involves a user at a DMC or an M-DMC, which enables the user to find content on a DMS that in turn will be played on a user selected DMR.

- 2-Box Printing System Usage

  This usage involves a user at a Printing Controller-1, which enables the user to set up image print tasks with a DMPr.

- 3-Box Printing System Usage

  This usage involves a user at a Printing Controller-2, which enables the user to find images on a DMS or M-DMS and then set up a print tasks with a DMPr.

- Download System Usage

  This usage involves a user at a Download Controller or an M-DMD, which enables the user to download content from a DMS or an M-DMS so that the Download Controller or the M-DMD has its own copy.

- Upload System Usage

  This usage involves a user at an Upload Controller or an M-DMU, which enables the user to send content to a DMS or an M-DMS with the Upload Device Option so that the DMS or the M-DMS can distribute the content to other endpoints.

- Upload Synchronization System Usage

  This usage involves a user at an Upload Synchronization Controller, which enables the user to reflect any changes to the local store of content into a DMS or an M-DMS with the Content Synchronization Device Option so that the DMS or the M-DMS can receive and distribute the new or changed content to other endpoints.

- Download Synchronization System Usage

  This usage involves a user at a Download Synchronization Controller, which enables the user to obtain any changes to the store of content on a DMS or an M-DMS supporting the Content Synchronization Device Option.

- 2 Box RUI Pull with/without A/V System Usage

  This usage involves a user at a RUI Pull Controller (+RUIPL+), which enables a user to find and interact with a user interface that is offered by a RUI Source (+RUISRC+), but which is rendered by the RUI Pull Controller. A user interface might control A/V content that is rendered inside the user interface.

- 3 Box UI-only System Usage

  This usage involves a user at a RUI Controller (+RUICTRL+), which enables a user to set up a remote UI connection between a RUI Sink (+RUISINK+) and a remote UI offered by a RUI Source (+RUISRC+). This system usage does not include control of A/V content that is rendered inside the user interface.

- 3 Box UI with A/V System Usage

  This usage involves a user at a RUI Controller (+RUICTRL+), which enables a user to set up a remote UI connection between a RUI Sink (+RUISINK+) and a remote UI offered by a RUI Source (+RUISRC+), that includes control of A/V content that is rendered inside the user interface.

- Scheduled Recording System Usage

  This usage involves a user at a Scheduled Recording Controller, which enables the user to instruct a media server (DMS/M-DMS) to browse, create, modify, and cancel scheduled recordings.

- EPG System Usage

  This usage involves a user at an EPG Controller, which enables the user to view EPG metadata exposed by a DMS or M-DMS.

Subclauses 5.7.1 through 5.7.14 will briefly describe each of the System Usages and their respective device interaction models. In these System Usages, devices in the MHD and HND Device Categories can achieve interoperability by utilizing an MIU with the responsibility for media format level conversions, and/or an M-NCF for bridging the networking connectivity layers between the two Device Categories. These elements are described in 5.8. For clarity, the device interaction model diagrams in 5.7.1 through 5.7.14 show HND and MHD devices interacting directly. All these System Usages are shown with Device Functions that are media transport agnostic. All of these System Usages imply HTTP as the mandatory media transport and other optional media transports, such as RTP, can be substituted.

5.7.1 2-Box Pull System Usage

The 2-Box Pull System Usage pulls DLNA compliant content from a media server (DMS/M-DMS) to be rendered locally by the device pulling the content (DMP, M-DMP). The user perspective is that content is being pulled to the DMP or the M-DMP for immediate rendering on the device. The user is browsing and selecting content on the DMS or the M-DMS. This usage between a DMS and a DMP was the only System Usage supported in the v1.0 Interoperability Guidelines. Note that the rendering function is not exposed onto the network in a DMP or an M-DMP implementation. Also note that in all of the following System Usage diagrams, the Media Transport Client/Server are for the media transport layer only. The UPnP Device/CP has HTTP functions independent of the media transport layer and is implied as being part of the UPnP Device/CP Device Functions. Figure 3 illustrates this device interaction model. The following steps are performed in this System Usage:

1. Invoke UPnP actions to browse and select content.
2. Request the content for playback.
3. Transport the content to the DMP or the M-DMP.

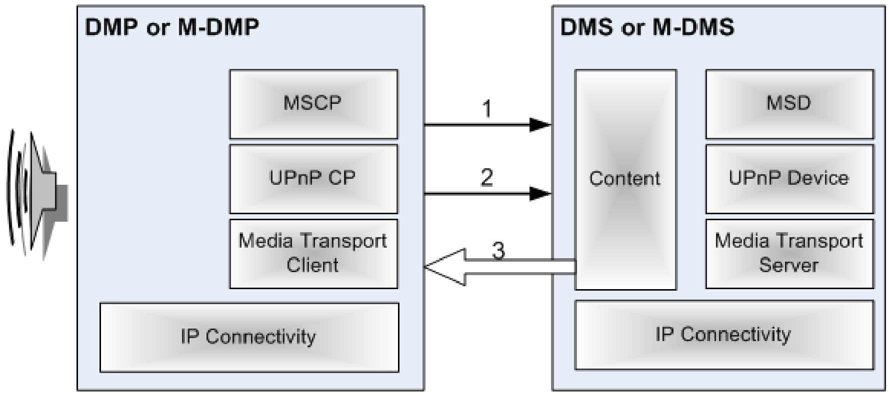


Figure 3 — 2-Box Pull System Usage Interaction Model

5.7.2 2-Box Push System Usage

The 2-Box Push System Usage pushes DLNA compliant content to a rendering device (DMR). The user perspective is that content is being pushed to the DMR even though content might actually be transported in a "pull" manner depending on the media transport used. The user is selecting content at the device where the content is resident. Figure 4 illustrates this device interaction model. The following steps are performed in this System Usage:

1. Invoke UPnP actions to set up a playback session.
2. Request the content for playback.
3. Transport the content to the DMR.

Note that the Push Controller Device Capability functionality can only be incorporated as part of any physical device with a valid DLNA Device Class. It shall never appear as a stand-alone device. This is how the Push Controller Device Capability inherits other Device Functions (e.g. IP Connectivity) at other layers in the DLNA Device Architecture. This is applicable to Device Classes in both the HND and MHD Device Categories.

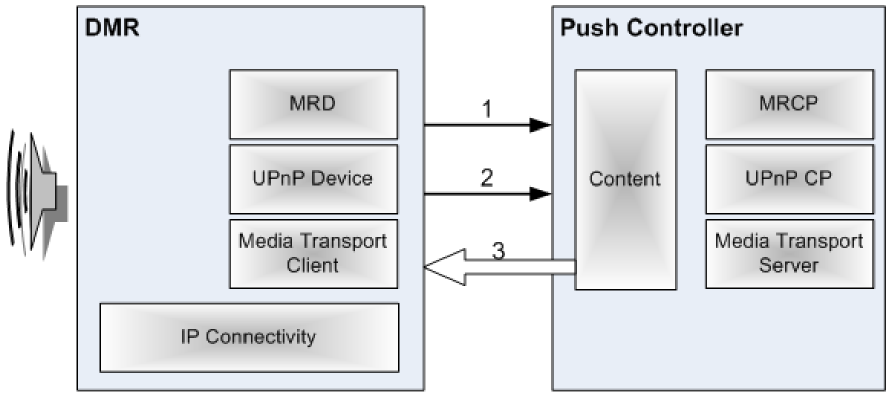


Figure 4 — 2-Box Push System Usage Interaction Model

5.7.3 3-Box System Usage

The 3-Box System Usage uses a device controller (DMC/M-DMC) to browse content on a media server (DMS/M-DMS) and to select a rendering device (DMR) to play the selected content. The DMC or the M-DMC is responsible for making sure a DMR can render the selected DLNA content.

Figure 5 illustrates this device interaction model. The following steps are performed in this System Usage:

1. Invoke UPnP actions to browse and select content.
2. Invoke UPnP actions to verify that the DMR has the capability to render the selected content and then set up a connection for the selected content between the DMR and the DMS or the M-DMS.
3. Request the content for playback.
4. Transport the content to the DMR.

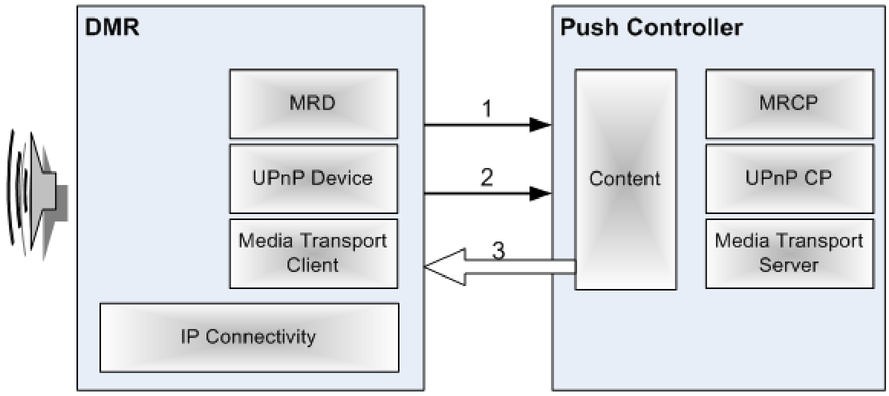


Figure 5 — 3-Box System Usage Interaction Model

5.7.4 2-Box Printing System Usage

The 2-Box Printing System Usage has a Printing Controller-1 that instructs an image printing device (DMPr) to print image(s) served locally by the Printing Controller-1. Print jobs are XHTML-Print documents that are sourced by the Printing Controller-1 and all images referenced in an XHTML-Print document are served up locally by the Printing Controller-1.

Figure 6 illustrates this device interaction model. The following steps are performed in this System Usage:

1. Invoke UPnP actions to the DMPr to initiate the print job by providing the URL to the XHTML-Print document.
2. The DMPr acquires the XHTML-Print document from the Printing Controller-1.
3. The DMPr requests the Image content from the Printing Controller 1.
4. The Printing Controller 1 responds with the Image content.

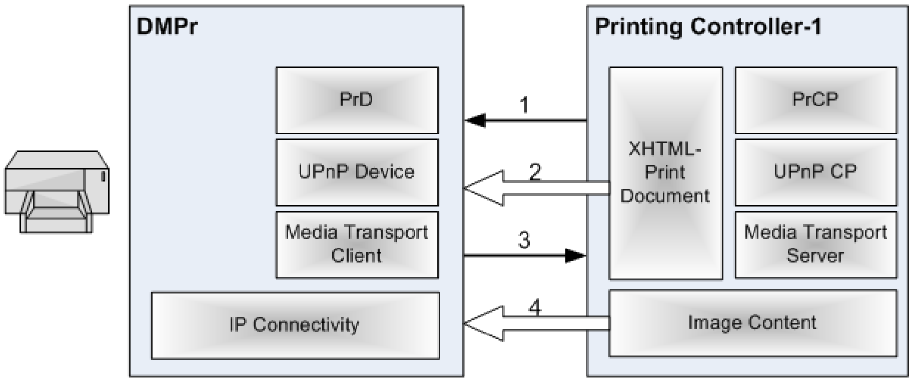


Figure 6 — 2-Box Printing System Usage Interaction Model

5.7.5 3-Box Printing System Usage

The 3-Box Printing System Usage has a Printing Controller-2 that instructs an image printing device (DMPr) to print image(s) served by a DMS or M-DMS Device Class. The Printing Controller-2 has the additional functionality for finding and selecting the image(s) exposed by a DMS and/or M-DMS. Print jobs are XHTML-Print documents that are sourced by the Printing Controller-2 and all images referenced in an XHTML-Print document are served up by a DMS and/or M-DMS.

Figure 7 illustrates this device interaction model. The following steps are performed in this System Usage:

1. Invoke UPnP actions to browse and select image content.
2. Get the URLs for the selected image content and reference the URLs in the XHTML-Print document.
3. Invoke UPnP actions on a DMPr to initiate the print job by providing the URL to the XHTML-Print document.
4. The DMPr acquires the XHTML-Print document from the Printing Controller-2.
5. The DMPr requests the Image content from the DMS or M-DMS.
6. The DMS or M-DMS responds with the Image content.

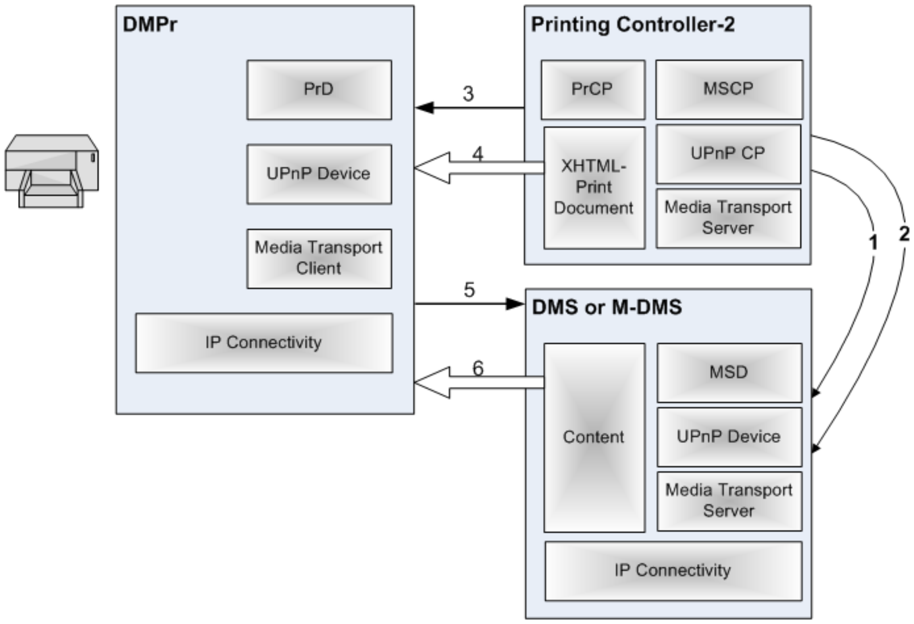


Figure 7 — 3-Box Printing System Usage Interaction Model

5.7.6 Download System Usage

The Download System Usage allows a Download Controller or an M-DMD to transfer and store DLNA content from a media server (DMS or M-DMS).

Figure 8 illustrates this device interaction model. The following steps are performed in this System Usage:

1. Invoke UPnP actions to find content to download.
2. Request the content that needs to be downloaded.
3. Transport content to the Download Controller or the M-DMD.

Note that the Download Controller Device Capability functionality can only be incorporated as part of any physical device with a valid DLNA Device Class. It can never appear as a stand-alone device. This is how the Download Controller Device Capability inherits other Device Functions (e.g. IP Connectivity) at other layers in the DLNA Device Architecture. In the MHD environment, this System Usage can be accomplished by a device with only this functionality, hence the need for an M-DMD Device Class to provide support for all layers in the DLNA architecture. This is not a requirement in the HND environment which shall incorporate this functionality as an addition to an existing Device Class.

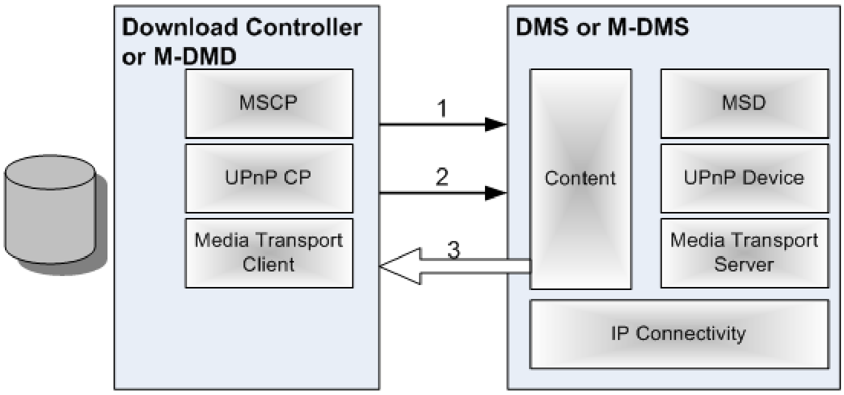


Figure 8 — Download System Usage Interaction Model

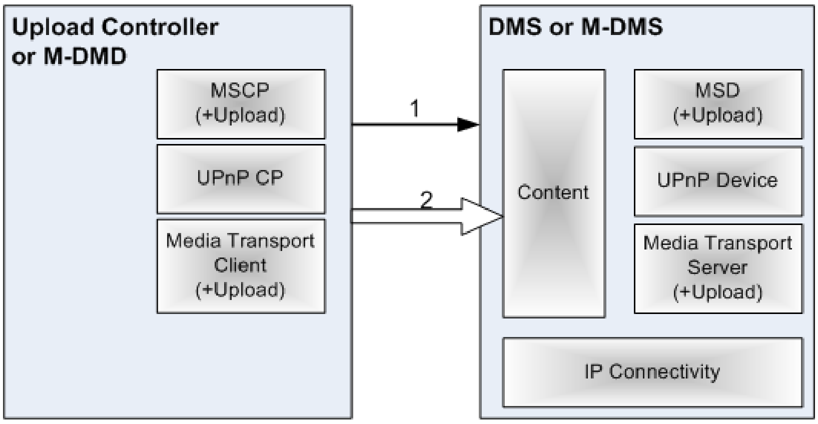
5.7.7 Upload System Usage

The Upload System Usage has an Upload Controller Device Capability or an M-DMU to instruct a media server (DMS/M-DMS) to accept some new content to be added to its list of available content.

Figure 9 illustrates this device interaction model. The following steps are performed in this System Usage:

1. Invoke UPnP actions to create a CDS entry for the content to be uploaded.
2. Transport the content being uploaded to the DMS or the M-DMS.

Note that the Upload Controller Device Capability functionality can only be incorporated as part of any physical device with a valid DLNA Device Class. It shall never appear as a stand-alone device. This is how the Upload Controller Device Capability inherits other Device Functions (e.g. IP Connectivity) at other layers in the DLNA Device Architecture. In the MHD environment, this System Usage can be accomplished by a device with only this functionality, hence the need for an M-DMU Device Class to provide support for all layers in the DLNA architecture. This is not a requirement in the HND environment which shall incorporate this functionality as an addition to an existing Device Class.



"+" Indicates additional functionality from v1.0 Interoperability Guidelines for a Device Function

Figure 9 — Upload System Usage Interaction Model

5.7.8 Download Synchronization System Usage

The Download Synchronization System Usage has a Download Synchronization Controller Device Capability receive changes in the content or metadata stored on a media server (DMS/M-DMS) and apply those changes to the local storage. Figure 8 Download Synchronization System Usage Interaction Model illustrates this device interaction model. The Media Server tracks changes within its metadata database and makes that information available to the controller. The controller decides what elements to download to the local storage. The following steps are performed in this System Usage:

1. The Download Synchronization Controller invokes UPnP actions to obtain a list of changes on the Media Server since the last synchronization.
2. The Download Synchronization Controller receives the resulting information from the Media Server on the changes that have occurred in the database.
3. The Download Synchronization Controller decides what actions to carry out to synchronize its local storage with the changes present on the Media Server.
4. The Download Synchronization Controller obtains the information necessary to download the required information (URLs and metadata)
5. If necessary the Download Synchronization Controller transfers the relevant content from the Media Server to the controller.

Note that the Download Synchronization Controller Device Capability functionality can only be incorporated as part of any physical device with a valid DLNA Device Class. It shall never appear as a stand-alone device. This is how the Download Synchronization Controller Device Capability inherits other Device Functions (e.g. IP Connectivity) at other layers in the DLNA Device Architecture.

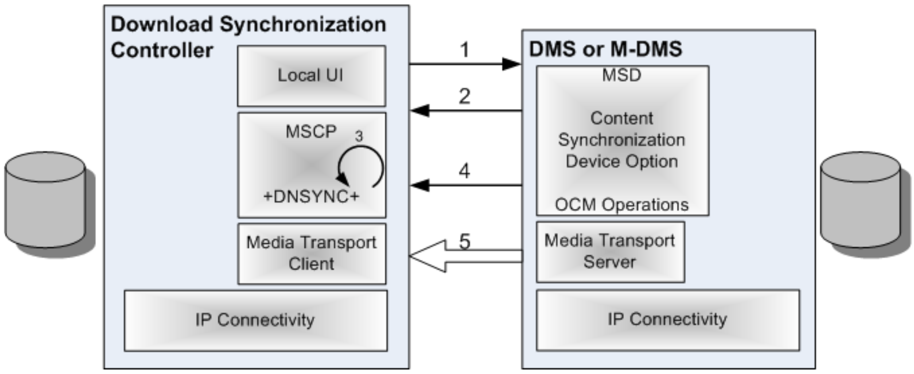


Figure 10 — Download Synchronization System Usage Interaction Model

5.7.9 Upload Synchronization System Usage

The Upload Synchronization System Usage has an Upload Synchronization Controller Device Capability propagate changes in the local content or metadata to a media server (DMS/M-DMS) to be added to its list of available content. Figure 9 Upload Synchronization System Usage Interaction Model illustrates this device interaction model. The Upload Synchronization Controller tracks changes in the local storage and uploads those changes to a DMS or M-DMS with the CDS Tracking Changes Option. The following steps are performed in this System Usage:

1. The Upload Synchronization Controller determines that changes have been made in the local content storage area
2. The Upload Synchronization Controller invokes UPnP actions on the Media Server to obtain a list of changes since the last synchronization.
3. The Upload Synchronization Controller receives the resulting information from the Media Server on the changes that have occurred in the CDS.
4. The Upload Synchronization Controller determines what changes need to occur on the DMS or M-DMS to properly represent the local changes.
5. The Upload Synchronization Controller performs OCM operations to reflect the local changes onto the Media Server
6. If necessary the Upload Synchronization Controller tranfers the relevant content from the controller to the Media Server.

Note that the Upload Synchronization Controller Device Capability functionality can only be incorporated as part of a physical device with a valid DLNA Device Class. It shall never appear as a stand-alone device. This is how the Upload Synchronization Controller Device Capability inherits other Device Functions (e.g. IP Connectivity) at other layers in the DLNA Device Architecture. Please note that the Upload Synchronization Controller Device Capability contains the ability to use the OCM operations on the server to carry out the actions required for synchronization.

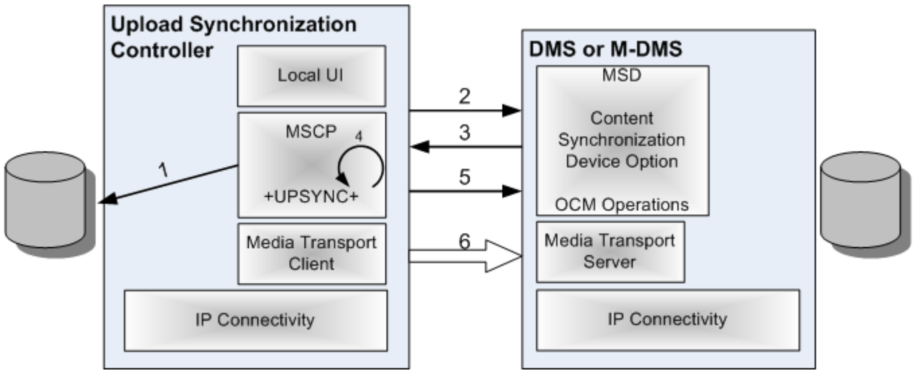


Figure 11 — Upload Synchronization System Usage Interaction Model

5.7.10 2 Box RUI Pull with/without A/V System Usage

This usage enables DLNA compliant remote UI content to be pulled from a RUI Source capability (+RUISRC+) in order to be rendered locally by a RUI Pull Controller (+RUIPL+).

Figure 12 illustrates this device interaction model without A/V. The following steps are performed in this System Usage:

1. Invoke actions to find remote UI content.
2. Request UI content.
3. Transport UI content to +RUIPL+.
4. Interaction between +RUIPL+ and +RUISRC+ via Remote UI connection.

Note that the +RUISRC+ capability includes a UPnP Device Function in order to make the capability discoverable in the network irrespective of the Device Class to which the capability is added. This means that capability can be added not only to discoverable Device Classes, but also to non-discoverable Device Classes, such as a (M-)DMC or (M-)DMP. There are no restrictions with which Device Class the +RUIPL+ and +RUISRC+ capabilities can be co-located, unless explicitly stated in the guidelines.

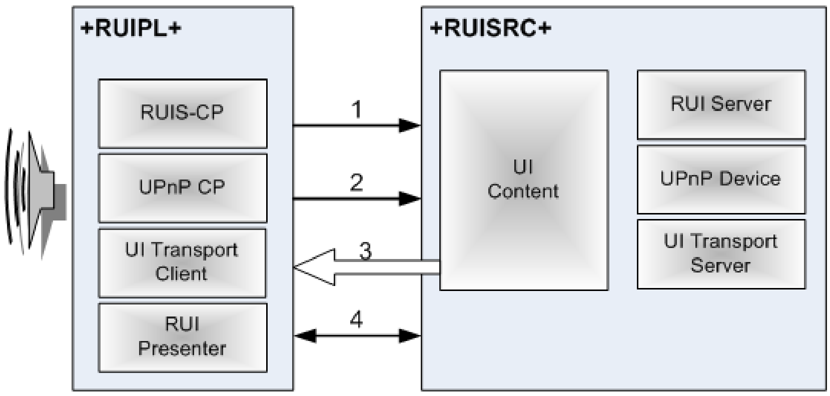


Figure 12 — RUI Pull without A/V System Usage Interaction Model

This System Usage can be extended to control, transport and render A/V content inside the UI. This is enabled through co-location of the +RUIPL+ and +RUISRC+ capability with the appropriate Media Transport Client/Server device functions of existing Device Classes to which the capabilities are added. Table 2 below shows the possibilities for co-location. Note: for the RUI Pull without A/V system usage as denoted above, there are no restrictions on co-location.

Table 2 — Collocation possibilities of +RUIPL+ and +RUISRC+ capabilities for A/V

Capability	Required A/V components for collocation	Existing Device Classes/Capabilities with the Required A/V components
+RUIPL+	• MT Client	• MR • (M-)DMP
+RUISRC+	• Content • MT Server	• (M-)DMS • +PU+ (Push Controller)

Graphically the RUI Pull system usage with A/V rendering and control can be denoted as follows, see Figure 13 below, whereby the +RUIPL+ and +RUISRC+ capabilities are shown together with a hosting device class, that will offer the appropriate Media Transport components.

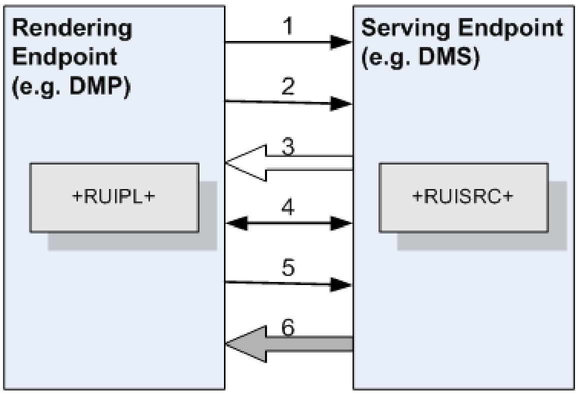


Figure 13 — RUI Pull with A/V System Usage Interaction Model

The +RUIPL+ and +RUISRC+ capabilities and steps 1 through 4 are the same as shown in Figure 12. Steps 5 and 6 are defined as follows:

5. Request associated A/V content
6. Transport the A/V content to the Rendering Endpoint

5.7.11 3 Box UI-only System Usage

This usage enables a RUI Controller (+RUICTRL+) to set up a connection between a RUI Sink (+RUISINK+) and a remote UI offered by a RUI Source (+RUISRC+). This only pertains to the initial setup phase from a remote UI point of view. After this point, it is unspecified where the user is located, i.e. for interacting with the remote UI and setting up the A/V. This can for example be done by using some means of user input on the +RUISINK+ or +RUISRC+ or by pairing the +RUICTRL+ and the +RUISINK+ using an out-of-band mechanism to provide user input (e.g. using infra-red). In the 3-Box UI-only case the System Usage does not include A/V content that is rendered as part of the user interface. Including A/V content in the 3-Box case is discussed in the next subclause (5.7.12).

Figure 14 illustrates this device interaction model. The following steps are performed in this System Usage:

1. Discover and match RUI.
2. Instruct +RUISINK+ to set up a remote UI connection to +RUISRC+
3. Request UI content
4. Transport UI content to +RUISINK+
5. Interaction between +RUISINK+ and +RUISRC+ via Remote UI connection

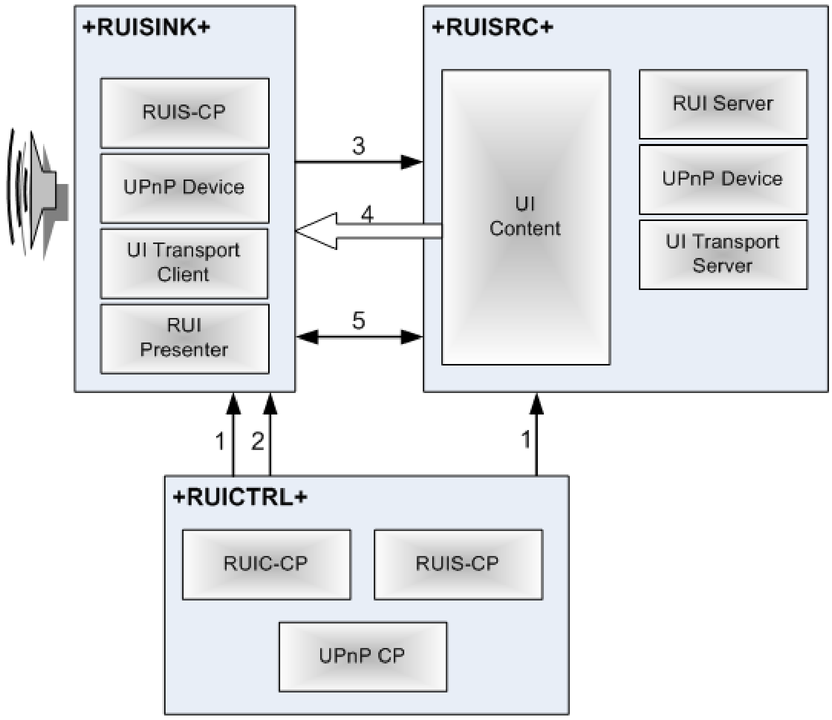


Figure 14 — 3-Box UI-only System Usage Interaction Model

Note that the 3-Box model actually refers to having three logical entities and not necessarily three physical entities. As shown in Figure 15 and Figure 16 below, it is possible to combine the logical components (i.e. +RUISINK+, +RUISRC+ and +RUICTRL+ capabilities) of the 3-box model into two physical entities. In this case, certain use cases (such as remote device configuration and setup) can be implemented either by using the 2-Box RUI Pull model defined in the previous subclauses, or by using such 3-Box model.

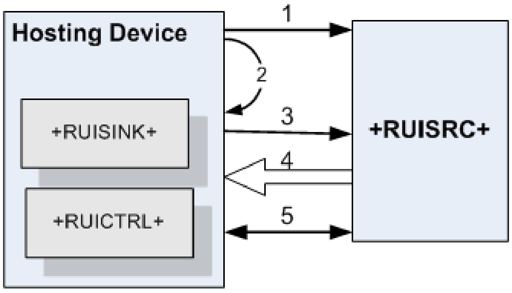


Figure 15 — Physical Box Configuration for 3-Box UI-only System Usage Model

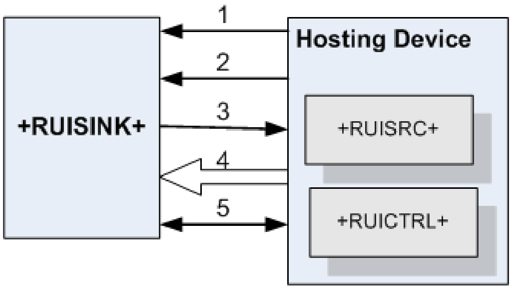


Figure 16 — Physical Box Configuration for 3-Box UI-only System Usage Model

As with any system usage it is possible to combine different system usages together, or use multiple instances of the same system usage to implement a certain use case. For example, the following figure describes a more complicated usage model which is derived from a use case whereby a mobile phone is interacting with a remote UI server, after which the remote UI gets transferred to a large TV set. This use case can be implemented either by combining a 2-Box RUI Pull with a 3-Box RUI model simultaneously, or by combining/using two instances of the 3-Box RUI model.

The following Figure 17 shows this use case by combining two instances of the 3-Box model.

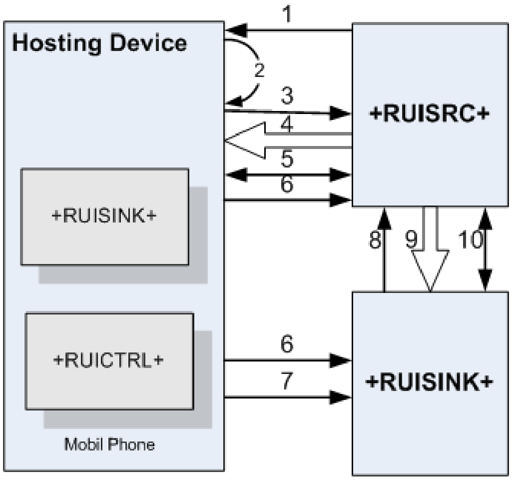


Figure 17 — Combining 2 Instances of 3-Box UI-only System Usage

This system usage model consists of the following steps:

1. Discover and match RUI.
2. Instruct +RUISINK+ to connect to +RUISRC+ (might be done internally, not via the network.
3. Request UI content from +RUISRC+.
4. Transport UI content to +RUISINK+.
5. Interaction between +RUISINK+ and +RUISRC+ via Remote UI connection.
6. Discover and match RUI.
7. Instruct +RUISINK+ to connect to +RUISRC+ (might be done internally, not via the network).
8. Request UI content from +RUISRC+.
9. Transport UI content to +RUISINK+.
10. Interaction between +RUISINK+ and +RUISRC+ via remote UI connection.

5.7.12 3 Box UI with A/V System Usage

Rendering A/V in the 3-Box Model is enabled through the co-location of the +RUISRC+ and +RUISINK+ Device Capabilities with existing Device Classes that offer the appropriate A/V serving and rendering components, in a similar way as is done for the 2-Box RUI Pull model with A/V (as defined in 5.7.10). If 3-box RUI needs to be combined with A/V, the following holds:

Table 3 — Collocation possibilities of +RUISRC+ and +RUISINK+ capabilities for A/V

Capability	Required A/V components for collocation	Existing Device Classes/Capabilities with the Required A/V components
+RUISRC+	• Content • MT Server	• (M-)DMS • +PU+ (Push Controller)

Capability	Required A/V components for collocation	Existing Device Classes/Capabilities with the Required A/V components
+RUISINK+	• MT Client	• MR • (M-)DMP

However, this only deals with a 2-Box like A/V model, while the RUI is using a 3-Box model. Another 3-Box case is whereby the A/V is set up by using a 3-Box model. This is for example the case when a proxy server is used to serve a remote UI for a non-remote UI enabled DMS device, whereby the remote UI is used to control A/V content coming from the DMS device. Figure 18 illustrates such device interaction model whereby a 3-Box Remote UI model is combined with a 3-Box A/V model. In this case, the +RUISINK+ capability is co-located with a DMR, and the +RUISRC+ capability is co-located with a (M-)DMC (could also be a DMS itself) that has a control point to discover a DMS in the network. The following steps are performed in this System Usage:

1. Discover and match RUI.
2. Instruct +RUISINK+ to connect to +RUISRC+ (might be done internally, not via the network if on same device)
3. Request UI content
4. Transport UI content to +RUISINK+
5. Interaction between +RUISINK+ and (M)-DMC via Remote UI connection
6. Browse and select A/V content
7. Invoke actions to setup connection between DMR and (M)-DMS
8. Request and control A/V content
9. Transport the A/V content to the Rendering Endpoint

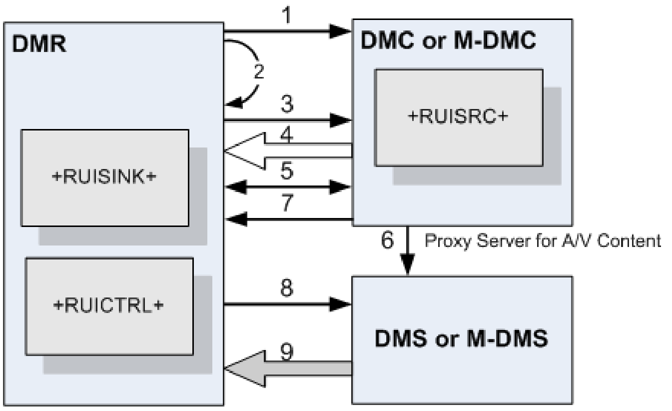


Figure 18 — 3-Box UI with 3-Box A/V System Usage Interaction Model

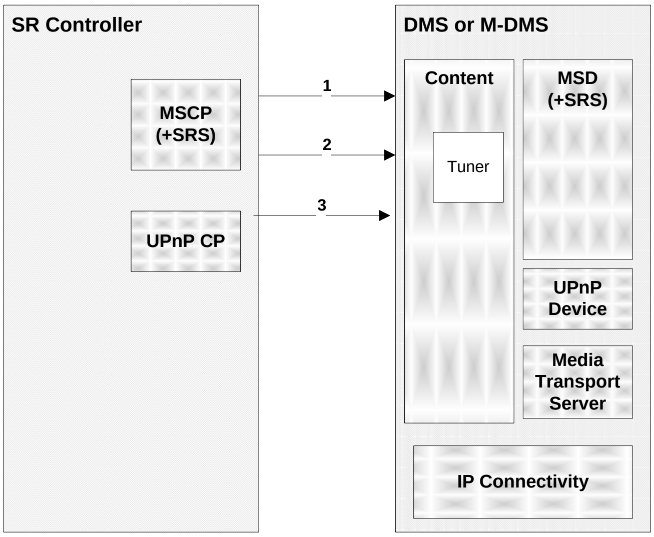
5.7.13 Scheduled Recording System Usage

The Scheduled Recording System Usage has a Scheduled Recording (SR) Controller Device Capability to instruct a media server (DMS/M-DMS) to create, modify, and cancel a scheduled recording. Figure 5-9 illustrates this device interaction model. The following steps are performed in this System Usage:

1. Invoke UPnP actions to obtain some or all of the values for the input parameters needed for setting up a scheduled recording.

2. Invoke UPnP actions to create a scheduled recording using the input parameters obtained in step 1.
3. Invoke UPnP actions to allow for the browsing, modification, and cancellation of existing scheduled recordings.

Note that the SR Controller Device Capability functionality can only be incorporated as part of any physical device with a valid DLNA Device Class. It shall never appear as a stand-alone device. This is how the SR Controller Device Capability inherits other Device Functions (e.g. IP Connectivity) at other layers in the DLNA Device Architecture. Also note that the UPnP Scheduled Recording Service Device Option is a UPnP service integrated into a UPnP MediaServer Device. The SR Controller includes a MSCP with functionality for scheduled recording.



"+" Indicates additional functionality from previous versions of the Interoperability Guidelines for a Device Function to implement this System Usage

Figure 19 — Scheduled Recording System Usage Interaction Model

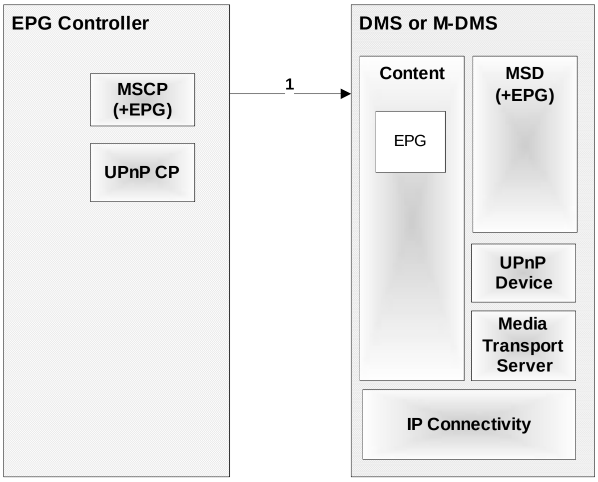
5.7.14 EPG System Usage

The EPG System Usage allows an EPG Controller to search EPG metadata exposed by a UPnP AV MediaServer (DMS/M-DMS). The EPG Controller Device Capability consists of a UPnP AV MediaServer control point with EPG client functionality. The following independent operations are performed in the System Usage:

1. Invoke UPnP actions to search EPG metadata in the CDS based on certain criteria
2. Optionally invoke UPnP actions to obtain channel line-up information

Note that the EPG Controller Device Capability functionality can only be incorporated as part of any physical device with a valid DLNA Device Class. It shall never appear as a stand-alone device. This is how the EPG Controller Device Capability inherits other Device Functions (e.g. IP Connectivity) at other layers in the DLNA Device Architecture.

To enable this System Usage the EPG Server Device Option shall be implemented in a DMS or an M-DMS. The EPG Server obtains EPG metadata from an external source, and maps a set of mandatory properties to the Server's Content Directory Service. Mapping definitions are provided for OpenEPG, TV-Anytime, and DVB-SI. Servers can choose to export rich data provided by an OpenEPG, or TV-Anytime service by exposing such XML based information as Foreign Metadata embedded in EPG Items in the CDS.



"+" Indicates additional functionality from previous versions of the Interoperability Guidelines for a Device Function to implement this System Usage

Figure 20 — EPG System Usage Interaction Model

5.8 Home Infrastructure Device (HID) System Usage

As described in 5.1, the HID Device Category contains Device Classes that enable interoperability for common System Usages between different Device Categories. In particular, the HND and MHD Device Categories share similar System Usages, but they differ in their Network Connectivity and Media Format requirements. The Device Class for bridging the Network Connectivity layer is the M-NCF Device Class. The Device Class for bridging the Media Format layer is the MIU Device Class. Both of these entities are described in this subclause.

Figure 21 illustrates these bridging functions for the 2-Box Pull System Usage between Device Classes in the MHD and HND Device Categories. The same concept can be applied to all of the System Usages described previously. In summary, network layer traffic is bridged by the M-NCF for communications between the MHD and HND Device Categories as needed and the MIU provides content transformation between the HND and MHD Device Categories as needed.

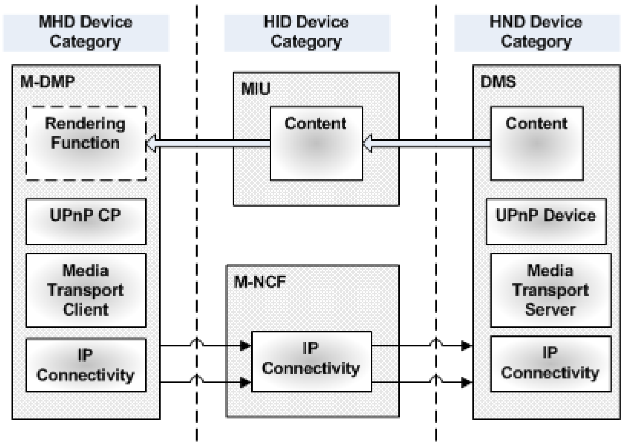


Figure 21 — 2-Box Pull System Usage Interaction Model Between Device Categories

5.8.1 Bridging HND and MHD Network Connectivity

Because devices in the HND and MHD Device Categories have different form factors, energy requirements and usages, they might support different link-level bearers to provide connectivity. For instance, it is common for MHD devices to support short-range and low-power consumption wireless bearers, such as Bluetooth. Furthermore, besides selection of bearers, MHD devices usually have different requirements than HND devices in two other areas at the Network Connectivity (NC) level, namely support for NC Power Savings and NC Security. For instance, different requirements in the area of NC Power Savings might be due to the fact that most MHD devices are battery operated, while in the area of NC Security to the fact that MHD devices might belong to visitors who only need to be given temporary access to the home network. To bridge the network connectivity gap between MHD and HND Device Categories that is created by possibly different bearers and different needs in NC Power Savings and NC Security, the Mobile Network Connectivity Function (M-NCF) Device Class is introduced. An example of the operation of the M-NCF is illustrated in Figure 22

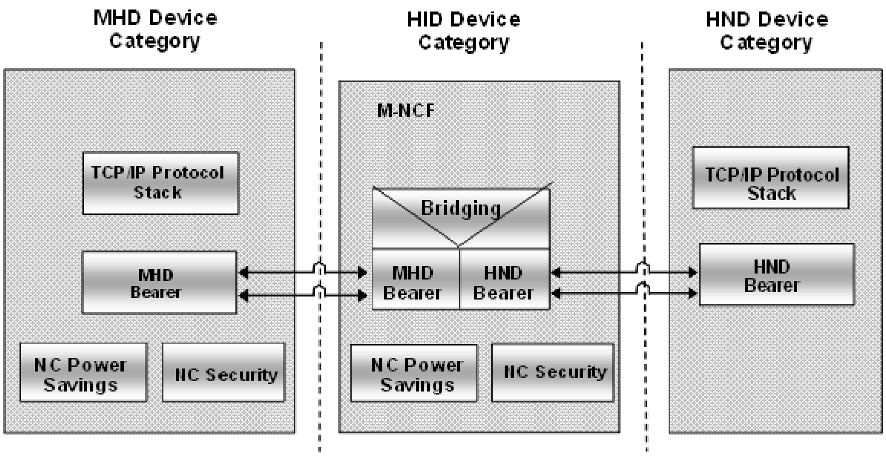


Figure 22 — M-NCF Bridging the Network Connectivity gap between MHD and HND

Devices in the MHD Device Category might support different bearers than those supported by devices in the HND category. It is the role of the M-NCF to do link-level bridging between different MHD and HND bearers. This link-level bridging function is transparent at the IP connectivity level and above. Furthermore, the M-NCF has the role of providing support for NC Power Savings to MHD devices that wish to conserve energy. This is done by leveraging the underlying power saving mechanisms provided by the MHD bearers and by, optionally, performing traffic reduction operations to prevent excess multicast and broadcast traffic from reaching the MHD devices. Finally, it is the role of the M-NCF to provide authenticated and encrypted access to MHD devices to the home network, whenever desired by its owner-administrator. To achieve that, the M-NCF provides required functionality to establish, manage and revoke the necessary permanent or temporary access rights for any MHD devices to be able to connect to the home network.

It is important to note that not all MHD devices are required to connect to the home network via an M-NCF. For example, an MHD device which supports a bearer which is also a HND bearer might connect to the home network without an M-NCF, in the same way that any HND device supporting this bearer would. However, whenever an M-NCF supporting the necessary MHD bearer exists, it is recommended that MHD devices connect to the home network via the M-NCF, to ensure better interoperability and better meeting the unique requirements of the MHD devices.

5.8.2 Bridging HND and MHD Media Formats

Because devices in the HND and MHD Device Categories have different form factors and usages, they also differ in the mandatory media format profiles which are supported. For instance, all HND devices that operate on audio/video data shall be able to process a profile based on MPEG2, while all MHD devices that operate on audio/video data shall be able to process a profile based on MPEG 4 Part 10. This leads to a potential lack of interoperability between devices of the different categories. The Media Interoperability Unit (MIU) is a Device Class of the HID Device Category that is defined to bridge this gap for the mandatory media format profiles and ensure media level interoperability between devices of the MHD and HND Device Categories. The operation of the MIU Device Class is shown in the following diagram.

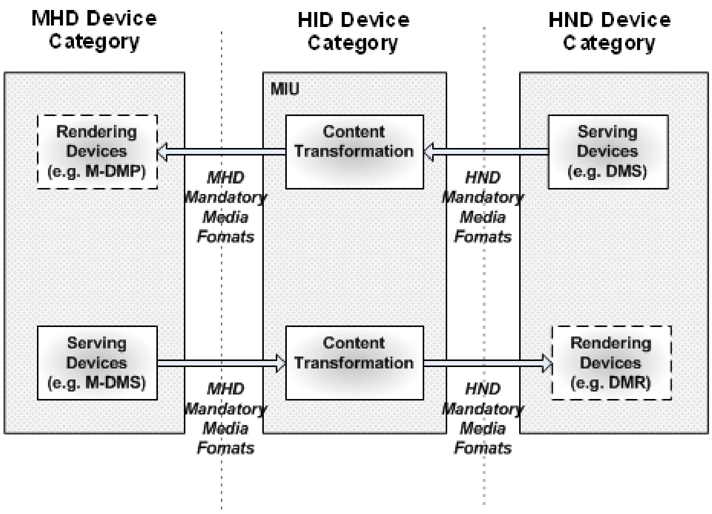


Figure 23 — Media Interoperability Between Device Categories

The devices of the MHD or HND Device Categories can serve or consume content in the mandatory media format profiles of the given Device Category. It is the role of the MIU to convert between the media format profiles so that content created and stored on an MHD device can be rendered on an HND device and vice versa.

It is important to note that the MIU is responsible only for media format level conversions. Because the core architecture of both the MHD and HND devices is the same set of UPnP devices, MHD devices are free to connect directly with their peer HND devices at any time, provided that they abide by the HND media format requirements. For example, MHD servers might find a DMR that can render their particular media format profiles, but they shall not expect that functionality from an HND device. In the case where the HND device does not directly meet the media requirements, the MHD device should connect to the MIU for the necessary services.

5.9 Interoperability Guidelines Usage

The guideline requirements tables found in clause 7 contain a column that specifies which Device Classes apply to a requirement. For the v1.0 Interoperability Guidelines, only DMS and DMP were applicable. For these Interoperability Guidelines, three new Device Classes are defined in addition to the two above for the HND Device Category. They are a DMC, DMR, and DMPr. The MHD Device Category with five new Device Classes is introduced in this version of the guidelines along with the two Device Classes of the HID Device Category. Table 4 summarizes all of the Device Classes in the HND Device Category and the mnemonics used within these Interoperability Guidelines. Table 3 summarizes all of the Device Capabilities that can be deployed with any Device Class and the mnemonics used within these Interoperability Guidelines. Table 6 summarizes all of the Device Classes in the MHD Device Category and the mnemonics used for these Device Classes. Table 7 contains the Device Classes in the HID Device Categories and the mnemonics used for these Device Classes.

Table 4 — DLNA Device Classes in the HND Device Category

DLNA Device Class	**Media Management Components**	**Media Transport Components**	**Functional Description**	**Device Classes or Capabilities Interacted with for Defined System Usages**	**Device Classes Interacted With Given Compatible Networking and Media Formats Profiles**
v1.0 Device Classes					
DMS (Digital Media Server)	MSD	Media Transport Server	Serves up media	DMP, DMC, DMR, DMPr, other endpoints with +UP+, +DN+, or +PR2+ capabilities	M-DMP, M-DMC, M-DMD, M-DMU
DMP (Digital Media Player)	MSCP	Media Transport Client	Selects, controls and renders the selected media	DMS	M-DMS
Device Classes new to v1.5					

DLNA Device Class	Media Management Components	Media Transport Components	Functional Description	Device Classes or Capabilities Interacted with for Defined System Usages	Device Classes Interacted With Given Compatible Networking and Media Formats Profiles
DMC (Digital Media Controller)	MSCP MRCP	n/a	Controls the content selection and content rendering between networked devices	DMS, DMR	M-DMS
DMR (Digital Media Renderer)	MRD	Media Transport Client	Renders content	DMC, DMS, other endpoints with +PU+ capabilities	M-DMC, M-DMS
DMPr (Digital Media Printer)	PrD	Media Transport Client	Prints images	DMS, other endpoints with +PR1+ or +PR2+ capabilities	M-DMS

A new concept introduced in this version of the Interoperability Guidelines is a Device Capability. A Device Capability can be applied to any valid DLNA Device Class from any Device Category. Device Capabilities inherit the IP Connectivity from the Device Category of the Device Classes it is combined with. There are no baseline (i.e. mandatory) Media Format interoperability requirements for Device Capabilities, unless otherwise specified by explicit guidelines. Table 5 summarizes all of the Device Capabilities used in the System Usages and the mnemonics used within these Interoperability Guidelines to specify which requirements apply to them.

Table 5 — DLNA Device Capabilities

DLNA Device Capability	Device Capability Controller Identifier	Media Management Components	Media Transport Components	Device Classes Interacted with for Defined System Usages
Push Controller	+PU+	MRCP	Media Transport Server	DMR
Printing Controller-1	+PR1+	PrCP	Media Transport Server	DMPr
Printing Controller-2	+PR2+	PrCP MSCP	Media Transport Server	DMPr, DMS, M-DMS
Download Controller	+DN+	MSCP	Media Transport Client	DMS. M-DMS

DLNA Device Capability	Device Capability Controller Identifier	Media Management Components	Media Transport Components	Device Classes Interacted with for Defined System Usages
Upload Controller	+UP+	MSCP	Media Transport Client	DMS, M-DMS
Upload Synchroni-zation Controller	+UPSYNC+	MSCP	Media Transport Client	DMS, M-DMS
Download Synchroni-zation Controller	+DNSYNC+	MSCP	Media Transport Client	DMS, M-DMS
RUI Pull Controller	+RUIPL+	See Table 2	See Table 2	+RUISRC+
RUI Source Capability	+RUISRC+	See Table 2	See Table 2	+RUIPL+ +RUISINK+ +RUICTRL+
RUI Sink Capability	+RUISINK+	See Table 3	See Table 3	+RUISRC+ +RUICTRL+
RUI Controller	+RUICTRL+	n/a	n/a	+RUISRC+ +RUISINK+
Scheduled Recording Controller	+SR+	MSCP	n/a	DMS, M-DMS
EPG Controller	+EPG+	MSCP	n/a	DMS, M-DMS

The MHD Device Category has different media format and network connectivity requirements because of various device constraints. Table 6 summarizes all of the Device Classes in the MHD Device Category and the mnemonics used within these Interoperability Guidelines.

Table 6 — DLNA Device Classes in the MHD Device Category

DLNA Device Class	Media Management Components	Media Transport Components	Functional Description	Device Classes Interacted with for Defined System Usages	Device Classes or Capabilities Interacted With Given Compatible Networking and Media Formats Profiles
Device Classes new to v1.5					
M-DMS (Mobile Digital Media Server)	MSD	Media Transport Server	Serves up media	M-DMP, M-DMC, M-DMD, M-DMU	DMP, DMC, DMR, DMPr, other endpoints with +UP+, +DN+, or +PR2+ capabilities

DLNA Device Class	Media Management Components	Media Transport Components	Functional Description	Device Classes Interacted with for Defined System Usages	Device Classes or Capabilities Interacted With Given Compatible Networking and Media Formats Profiles
M-DMP (Mobile Digital Media Player)	MSCP	Media Transport Client	Selects, controls and renders the selected media	M-DMS	DMS
M-DMC (Mobile Digital Media Controller)	MSCP MRCP	n/a	Controls the content selection and content rendering between networked devices	M-DMS, DMR	DMS
M-DMU (Mobile Digital Media Uploader)	MSCP	Media Transport Client	Uploads the selected media to servers	M-DMS	DMS
M-DMD (Mobile Digital Media Downloader)	MSCP	Media Transport Client	Selects, controls and downloads the selected media	M-DMS	DMS

Due to the differences in the Media Format Profile support and network connectivity requirements, the interoperability for Device Classes is only assured within a Device Category. To extend interoperability for the Device Classes in the MHD Device Category throughout the home network, including the Device Classes in the HND Device Category, a special Device Category named Home Infrastructure Device (HID) is defined. The HID Device Category ensures interoperability between devices of different categories and ensures interoperability between the Device Classes and capabilities and the devices listed in the final two columns in the above tables. Table 7 summarizes all of the Device Classes in the HID Device Category and the mnemonics used within these Interoperability Guideline

Table 7 — DLNA Device Classes in the HID Device Category

DLNA Device Class	Media Management Components	Media Transport Components	Functional Description
M-NCF (Mobile Network Connectivity Function)	n/a	n/a	Provides a network connectivity bridge between devices in the HND and MHD Device Categories.
MIU (Media Interoperability Unit)	MSD, MRD, MSCP, MRCP	Media Transport Server,Media Transport Client	Provides virtual services for content transformation between required media formats for devices in the HND and MHD Device Categories

6 Guideline Terminology and Conventions

6.1 Guideline Compliance Classifiers

Reference [71] provides a description of terminology conventions used in all IETF RFC documents. The terminology and conventions used by the DLNA Home Networked Device Interoperability Guidelines are adapted from this reference. The details of each guideline will carry a compliance classifier from the following set:

[M] Required, Shall: This is the minimum set of requirements that will ensure interoperability and/or robust operation between devices. All devices are expected to comply with these requirements when expressed in unconditional form. A conditional requirement expressed in the form, "If X, then Y shall be implemented", means that the requirement "Y" shall be met when the conditional aspect "X" applies to a given implementation.

[S]hould, Recommended: Recommended items are optional items that are strongly recommended for inclusion in products. The difference between "recommended" items and "optional" items, below, is one of priority. When considering features for inclusion in a product, recommended items should be included first.

[O]ptional, May: Optional items are suggestions for features that will enhance the user experience or are offered as a less preferred choice relative to another recommended feature. If optional features are included, they shall comply with the requirement to ensure interoperability with other implementations.

6.2 Standard or Specification Usage Classifiers

When specifying guideline details, it is often useful to reiterate or clarify certain aspects of a standard or specification that are often violated or misunderstood. Furthermore, there might be guidelines that intentionally contradict or restrict implementation of certain aspects of a standard or specification in order to ensure interoperability between DLNA devices. The following classifiers are used in the DLNA Home Networked Device Interoperability Guidelines to indicate the relationship of a specific guideline to a source standard or specification:

[A]dding: A guideline that adds to or supplements a standard or specification to enhance interoperability. A guideline that does not reference a standard or specification also uses this classifier.

[C]larifying: A guideline that addresses vague or ambiguous aspects of a standard or specification.

[F]ixing: A guideline that intentionally supersedes and fixes aspects of a standard or specification that is incorrect and would otherwise provide a poor user experience or prevent device interoperability.

[L]imiting: A guideline that narrows or specifies an exact behavior in areas where a standard or specification provides for greater degrees of latitude in implementation.

[R]epeating: A guideline that repeats what is already in a standard or specification because of observed and repeated problems with implementations. Whenever a guideline with this usage classifier seems to be in conflict with the actual standard, the standard prevails over the guideline.

6.3 Guideline Font Usage Conventions

The following font usage conventions are used within the DLNA Home Networked Device Interoperability Guidelines to provide additional clarity:

- Hyperlinks to reference citations are indicated as [number]. For example [1], [20], …
- UPnP action names are indicated as: [Service acronym]:[action name], such as CDS:Browse.
- Special terms are sometimes *italicized*. Sometimes a guideline will define a term for use within that guideline and the term will be *italicized*.

6.4 Guideline Syntax Notation Conventions

The following are syntax (BNF) notation conventions used within the DLNA Home Networked Device Interoperability Guidelines to provide readability.

- Linear whitespace (LWS) characters, such as carriage returns, spaces, tabs, or line feeds, are not implied anywhere in any of the syntax (BNF) definitions used within the Interoperability Guidelines.
- The use of LWS characters is restricted within the DLNA Interoperability Guideline unless explicitly specified in any of the syntax definitions with reference to UPnP HTTP communications.
- By default, text tokens and values have a case-sensitive treatment unless explicitly noted in the guidelines. This convention also applies to BNF definitions, XML tag names, XML tag values, capability IDs, and HTTP header values for HTTP headers used by the DLNA guidelines. One of the exceptions to this rule applies to the names of HTTP headers. HTTP header names have a case-insensitive treatment. For example TimeSeekRange.dlna.org is the same as timeseekrange.dlna.org. (See 7.5.4.3.2.6)). Other exceptions are described in each guidelines which define BNF syntax.

6.5 Guideline Normative and Informative Text Conventions

All text that appears in the DLNA Interoperability Guidelines is to be considered normative unless explicitly stated otherwise, such as informative references and informative appendices. Normative text includes text before guideline attributes tables, but testable guidelines are only contained within subclauses guideline attribute tables..

6.6 DLNA XML Namespaces & Schemas

The DLNA Interoperability Guidelines make numerous references to XML elements and attributes that are defined for DLNA Device Classes and Device Capabilities. However, these namespaces are intentionally not defined through a formal DLNA XML schema. This allows the DLNA Interoperability Guidelines to define new XML elements and attributes in the future, without having to define a new namespace or schema definition. DLNA Devices Classes and Device Capabilities are expected to exhibit tolerant behavior when encountering XML elements or attributes that are defined in the future, as required by existing guidelines 7.3.2.23 and 7.4.1.3.1. The following table lists the namespace values that are used by DLNA guidelines and the context for their usage.

Table 8 — DLNA Namespace Values

Namespace value	Usage Context
urn:schemas-dlna-org:device-1-0	Used for XML elements and attributes defined by DLNA Interoperability Guidelines for use in UPnP device description files.
urn:schemas-dlna-org:metadata-1-0/	Used for XML elements and attributes defined by DLNA Interoperability Guidelines for use in DIDL-Lite documents and fragments.

6.7 General Rules on XML Documents and Fragments

The DLNA Interoperability Guidelines use XML documents and fragments in UPnP communications. To clarify the responsibility of each DLNA Device Class and DLNA Device Capability, this subclause specifies the following general rules for XML documents and fragments;

- DLNA Device Classes and DLNA Device Capabilities that source XML documents or fragments have the responsibility to provide valid (semantically and syntactically correct) and well-formed XML documents and fragments. This also includes DLNA Device Classes and DLNA Device Capabilities that receive XML documents or fragments and modify them syntactically, semantically, or both.

- DLNA Device Classes and DLNA Device Capabilities that receive XML documents or fragments from DLNA Device Classes and DLNA Device Capabilities can assume that the received XML documents or fragments are valid and can forward them to other devices without validation.

7 Guideline Requirements

7.1 Guidelines Overview

7.1.1 General

Clause 7 covers the guidelines that enable vendors to build interoperable products. Devices built to the DLNA Home Networked Device Interoperability Guidelines will be able to manage, transfer, and play personal media over a home network.

These guidelines are in a clause/subclause format as shown in Figure 24.

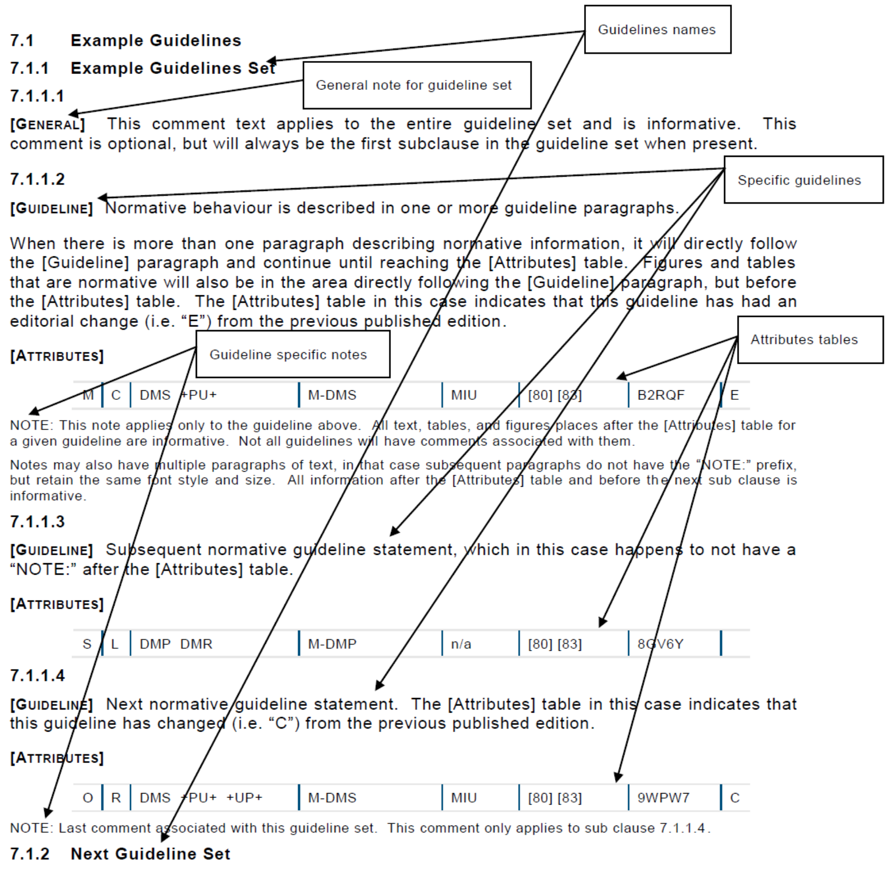


Figure 24 — Guideline Layout and Definitions

The following list describes the content of Figure 24:

a) Name: A label for the guideline set. The label is preceded with a sequentially increasing number to allow easy lookup.

b) Guidelines: The actual normative text of a guideline. A guideline is preceded with a sequentially increasing number in each part to allow easy lookup and the beginning of the paragraph starts with “[Guideline]”.

c) Attribute table: A summary of the essential attributes of a guideline. The table is preceeded with the paragraph text “[Attributes]” and is a single row with the following definitions for the columns:

- Compliance classifier: M/S/O (See 6.1 for the definition of guideline compliance classifiers).
- The specification usage classifier: A/C/F/L/R: for the guideline. (See 6.2 for the definition of specification usage classifiers.)
- HND Device Classes and Device Capabilities (see Table 4 and Table 5 for definitions). Device Capabilities are always listed in the HND column of the attribute table. Device Capabilities can also apply equally to the MHD Device Category but have been omitted from the MHD column in the attribute table to provide for better readability.
- MHD Device Classes(see Table 6 for definitions)
- HID Device Classes(see Table 7 for definitions)
- Standards citation: Standards that are referenced by the guideline. Standards citations are by number and are declared in Clause 2 and Annex M.
- Guideline unique number: an alpha-numeric string that uniquely identifies a guideline in all parts of this standard.
- Change indicator: documents the change in the guideline that occurred since the last edition of the Guideline (see Table 9 for definitions).

  Guideline attribute columns that do not have a value have the designation "n/a" (not applicable) . A visual map of possible values for the attribute tables is in Figure 25

d) Notes: Supplementary informative information about a guideline such as a justification for the guideline, the specific interoperability issue that is addressed, etc. The first paragraph in a note is preceeded with the text “NOTE:”.

Note that many guidelines do not explicitly list MIU since guidelines which apply to a device class also apply to the virtualized variants.

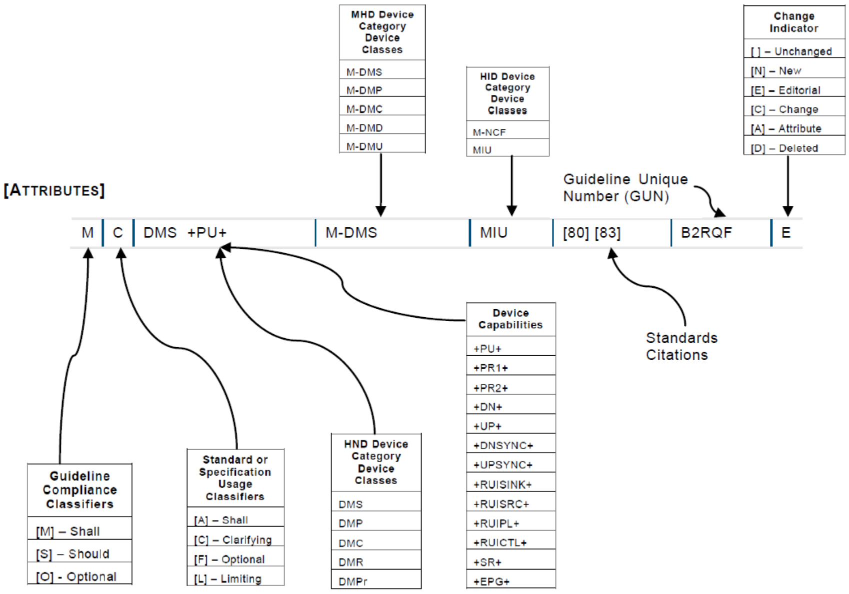


Figure 25 — Visual map of possible values for the attribute tables

Table 9 — Allowed Values for Change Indicator field in Attribute Table

Value	Meaning
<blank>	No changes in the text or figures from the immediately-preceding version of this guideline
A	Attribute table itself, excluding the change indicator, has changed. E.g. a new device class was added
C	changes made to the guideline modify the testing, intent, or other normative behavior relative to the immediately-preceding version of this guideline
D	Guideline has been deleted.
E	Changes made that do not modify testable guideline, intent, or other normative behavior
N	New. Guideline did not exist in previous Guideline versions

7.1.2 Conditions for Measuring Time in Message Exchanges

These guidelines define in certain cases time constraints for the exchange of messages between two communicating endpoints. These time constraints have been defined as a means to provide some operational consistency between the two communicating endpoints. However, in best-effort networks, actual time measurements for exchanging messages show wide variations depending on perturbations derived from network conditions, traffic, available bandwidth, and others. For this reason, this Subclause includes recommendations for conditions at the time of making these measurements:

- The two communicating endpoints should establish communications under Ideal Network Conditions.
- Time measurements at a given layer assume that the underlying layers preserve the communication channel. For example, time measurements at the HTTP layer cannot be valid if the underlying TCP/IP channel breaks during the measurements.
- Unless specified otherwise, time measurements assume that the communicating devices are both in active mode. This means that time measurements should not include transitions from sleep mode to active mode.

7.2 Networking and Connectivity

7.2.1 General

Networking and connectivity between devices is fundamental to the DLNA Home Networked Device Interoperability Guidelines. The family of protocols known as the Internet Protocol (IP) is the backbone for home network connectivity. Clusters of devices in the home can use other interconnect technologies, but IP ties these clusters together within the home, and provides connectivity outside the home to the global Internet. IP is independent of physical media and therefore there are a variety of connectivity options for DLNA devices.

The Networking and Connectivity guidelienes are organized in the following subclauses:

- Normative Definitions of NC-PS Modes
- Networking and Connectivity General Capability Requirements
- Networking and Connectivity QoS Requirements
- Networking and Connectivity Device Requirements

7.2.2 Normative Definitions of NC-PS Modes

Mobile Handheld Devices are typically battery powered and therefore saving power is a very important topic. Typically, these devices utilize power saving mechanisms both internally (e.g. place the processor in a low power state, turn-off the screen, etc.) and at the connectivity level to reduce their power utilization and extend their battery life. The NC-PS modes refer to power savings on the connectivity link between a Mobile Handheld Device and an M-NCF. Internal power saving measures are beyond the scope of the NC-PS guidelines. The NC-PS mode definitions are shown in Table 10. The definitions in Table 10 are normative because normative DLNA guidelines make reference to these NC-PS modes. The NC-PS modes are only defined for Bluetooth in this revision of the guidelines.

Table 10 — Normative Definitions of Network Connectivity Power Saving (NC-PS) Modes

NC-PS Mode	Definition
Active	The connection between a Mobile Handheld Device and an M-NCF is in the 'Active' mode, when the link-level connection has been established and no power saving mechanism is used in the underlying bearer used to transfer TCP/IP protocol-stack traffic. The Mobile Handheld Device has full IP connectivity at the highest possible data-rate and least latency.
Standby	The connection between a Mobile Handheld Device and an M-NCF is in the 'Standby' mode, when the link-level connection has been established and both the Mobile Handheld Device and the M-NCF have collaborated to put the link of the underlying bearer in a lower power state, using the mechanisms supported by that bearer. The Mobile Handheld Device still has full IP connectivity, but data-rates and latency can be affected adversely.

NC-PS Mode	Definition
Disconnected	The connection between a Mobile Handheld Device and an M-NCF is in the 'Disconnected' mode, when the link-level connection of the underlying bearer is disconnected. The Mobile Handheld Device has no IP connectivity.

7.2.3 Networking and Connectivity: General Capability Requirements

7.2.3.1 General

The guidelines in this subclause provide requirements for general capabilities. For example, these requirements describe the baseline capabilities of any Ethernet or Bluetooth implementation.

7.2.3.2 General Capability Requirements for Ethernet

7.2.3.2.1 NC Ethernet: Base

[GUIDELINE] If Ethernet is supported, IEEE 802.3i (10BASE-T) and 802.3u (100BASE TX) with auto negotiation capability and a connection to the network provided by an RJ45 connector is required.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[3]	2B2RQ	

7.2.3.2.2 NC Ethernet: Cabling

[GUIDELINE] If Ethernet is supported, any supplied network cabling should have a rating of Category 5e or better.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[87]	I8GV6	

7.2.3.2.3 NC Ethernet: Gigabit

[GUIDELINE] If Ethernet is supported, IEEE 802.3ab (1000BASE T) is recommended in addition to 7.2.3.2.1. An implementation shall support auto negotiation of gigabit operation with a similarly capable link partner and drop down to a lower speed as appropriate.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[3]	79WPW	

NOTE: Gigabit Ethernet is becoming available and affordable for home networks.

7.2.3.2.4 NC Ethernet: QoS Tolerance

[GUIDELINE] If Ethernet is supported, incoming tagged packets shall be tolerated. Tagged packets are Ethernet packets that include priority tags conformant with [3] 3.5, entitled 'Elements of the Tagged MAC Frame'. Here, 'tolerate' means that the packet payload of any received tagged or untagged packet shall be properly passed up to the higher layers in the network stack.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[3]	U7GZE	

NOTE: Packet tagging is the only QoS mechanism available on Ethernet at the link layer. Many devices on home networks are already capable of sending tagged frames, so all devices need to be able to tolerate them. For guidelines on tagging, see 7.2.4.2.1 and 7.2.4.2.2.

7.2.3.3 General Capability Requirements for 802.11

7.2.3.3.1 NC 802.11: Base

7.2.3.3.1.1

[GUIDELINE] If 802.11 is supported, one or more of the following radio selections is allowed:

- 802.11a
- 802.11b
- 802.11g
- 802.11n
- Wi-Fi Direct

For example, 802.11a, 802.11b, 802.11g, 802.11a/b, 802.11b/g, and 802.11a/b/g all meet this requirement.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[84] [100] [103]	T4RMV	C

NOTE: There is no implied requirement that a device needs to support multiple radios nor is it prohibited.

See Annex A for recommendations on Wireless Access Points and how they will help enable interoperability between products with different radio selections.

7.2.3.3.1.2

[GUIDELINE] If 802.11 is supported, the implementation shall support infrastructure mode operation..

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[84] [100] [103]	PB3QQ	C

NOTE: Some DLNA Device Classes might be required to support Ad-hoc (IBSS) mode for Wi-Fi conformance, however, the Interoperability Guidelines do not provide any requirements for Ad-hoc (IBSS) operation. Devices can assume infrastructure mode as the default.

7.2.3.3.2 NC 802.11: Wi-Fi and Wi-Fi Protected Setup Conformance

7.2.3.3.2.1

[GUIDELINE] If an 802.11 radio interface is supported, the implementation shall conform to one or more of the WFA test plans for 802.11 a/b/g [84], 802.11n [100], or Wi-Fi Direct [103]at the time the product is offered to the market.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[84] [100] [103]	YSXRI	C

NOTE: WFA is the industry consortium that does 802.11 compatibility testing. Wi-Fi interoperability requirements are increasing with time as new capabilities and features are specified by IEEE 802.11. When these capabilities are added to the Wi-Fi certification test plans, wireless implementations will have to conform to them.

Wi-Fi Direct devices that pass the P2P System Interoperability Test Plan [103] will pass one or more of the following: 802.11 a/g Interoperability Test Plan [84], or 802.11n System Interoperability Test Plan [100]. Additional Wi-Fi Direct perquisites include Wi-Fi WSC Test Plan [99] and WMM Test Plan [86]. 802.11b is not supported by Wi-Fi Direct.

7.2.3.3.2.2

[GUIDELINE] If Wi-Fi Direct is a supported radio interface, then the device will be WFA Wi-Fi Direct certified to support Intra-BSS Distribution.

[ATTRIBUTES]

M	A	n/a	n/a	n/a	[103]	JWF45	N

NOTE: Wi-Fi Intra-BSS Distribution is the name of the feature for bridging between members of the group.

7.2.3.3.2.3

[GUIDELINE] If Wi-Fi Simple Config is supported, the implementation shall conform to Wi-Fi Simple Config test plan requirements at the time the product is offered to the market.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[99]	X4RS9	C

7.2.3.4 General Capability Requirements: Bluetooth

7.2.3.4.1 NC Bluetooth: Base

7.2.3.4.1.1

[GUIDELINE] If Bluetooth is supported, devices shall implement either Bluetooth 1.1 or 1.2. In addition, devices shall implement PAN profile 1.0..

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[81] [82]	9WQ5N	

NOTE: The PAN profile provides IP connectivity over Bluetooth.

7.2.3.4.1.2

[GUIDELINE] If Bluetooth is supported, devices should implement Bluetooth 1.2.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[81] [83]	WS8V6	

NOTE: Bluetooth version 1.2 provides features such as auto-detecting 802.11; therefore this would limit interference for 802.11.

7.2.3.4.2 NC Bluetooth: Baseband Multi-slot Operation

7.2.3.4.2.1

[GUIDELINE] If Bluetooth is supported, Bluetooth multi-slot packet types shall be supported. Supported Bluetooth packet types are DM1, DH1, DM3, DH3, DM5 and DH5.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[82] [83]	OQ46A	

NOTE: Data rate can be increased by aggregating timeslots during transmission. Utilization of such mechanisms to improve user experience is not specified by the guidelines and is left to the vendors.

7.2.3.4.3 NC Bluetooth: LMP Support

7.2.3.4.3.1

[GUIDELINE] If Bluetooth is supported, LMP Power Control message types shall be supported

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[82] [83]	5G8VU	

NOTE: LMP Power Control message types are

LMP_incr_power_req

LMP_decr_power_req

LMP_max_power

LMP_min_power

7.2.3.4.3.2

[GUIDELINE] If Bluetooth is supported, the full range of 'Sniff' LMP parameters shall be supported.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[82]	TQLFG	

NOTE: This will ensure that the Bluetooth link will be able to enter the 'Sniff' mode with the parameters requested by a MHD, as specified in these guidelines.

7.2.3.4.4 NC Bluetooth: Connection Establishment

7.2.3.4.4.1

[GENERAL] The PAN profile specifies the process of Bluetooth connection establishment between a PANU (MHD) and a NAP (M-NCF). Guidelines on M-NCF and MHD device requirements speficy the M-NCF to be a NAP and the MHD device as a PANU.

7.2.3.4.4.2

[GUIDELINE] If Bluetooth is supported, the NAP and PANU shall co-operate to establish a Bluetooth connection as specified by the PAN profile.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[80] [82]	B2RQF	

7.2.3.4.4.3

[GUIDELINE] If Bluetooth is supported, the PANU device should initiate the Bluetooth connection to the NAP.

[ATTRIBUTES]

S	L	n/a	n/a	n/a	[80] [82]	8GV6Y	

7.2.3.4.4.4

[GUIDELINE] If Bluetooth is supported, the NAP may initiate a connection to a device that advertises a PANU service record.

[ATTRIBUTES]

O	L	n/a	n/a	n/a	[80] [82]	9WPW7	

7.2.3.4.5 NC Bluetooth: Bluetooth security mode

7.2.3.4.5.1

[GENERAL] Bluetooth Security Modes:

a) Non-secure

b) Service-level enforced security

c) Link-level enforced security

PAN profile service level enforced security mode consists of PAN Profile Authorization mode & PAN profile secrecy mode.

Recommendation of use could be reflected in the selection of device default values.

7.2.3.4.5.2

[GUIDELINE] If Bluetooth is supported, Bluetooth Security Mode 2 (service-level enforced security) shall be supported..

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[80] [82]	7GZEE

7.2.3.4.5.3

[GUIDELINE] If Bluetooth is supported, Bluetooth Security Mode 2 (service-level enforced security) should be used..

[ATTRIBUTES]

S	L	n/a	n/a	n/a	[80] [82]	4RMVW

7.2.3.4.6 NC Bluetooth: Bluetooth Authentication and Pairing

7.2.3.4.6.1

[GUIDELINE] If Bluetooth is supported, when device authentication is necessary, the Bluetooth authentication procedure as specified in [82] Part C subclause 3.2 shall be used.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[82]	B3QQ4

7.2.3.4.6.2

[GUIDELINE] If Bluetooth is supported, when device pairing is necessary, Bluetooth pairing procedure as specified in [82] Part C Subclause 3.3 shall be used.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[82]	SXRIP

7.2.3.4.6.3

[GUIDELINE] If Bluetooth is supported, the "pairable" mode shall be supported.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[82]	WQ5NZ

7.2.3.4.6.4

[Guideline] If Bluetooth is supported, the Bluetooth PIN code used for pairing shall contain only numeric characters: "0-9".

[Attributes]

M	L	n/a	n/a	n/a	[82]	S8V64	

NOTE: Some devices do not support non-numeric PIN codes.

7.2.3.4.7 NC Bluetooth: Bluetooth PAN Authentication

7.2.3.4.7.1

[General] PAN Profile Authorization Modes:

a) Open PAN

b) Authentication required

c) Authorization and authentication required

Recommendation is that manufacturers should enable security features by default. Higher layer security mechanisms e.g. 802.1x or others may also be used as described in BT PAN specification.

7.2.3.4.7.2

[Guideline] If Bluetooth is supported, then the PAN Profile v1.0 Authorization Mode 3 (authentication and authorization required) at the Bluetooth level shall be supported.

[Attributes]

M	R	n/a	n/a	n/a	[80] [82]	Q46AZ	

7.2.3.4.7.3

[Guideline] If Bluetooth is supported, then the PAN Profile v1.0 Authorization Mode 3 (authentication and authorization required) at the Bluetooth level should be used when connecting.

[Attributes]

S	L	n/a	n/a	n/a	[80] [82]	G8VUO	

7.2.3.4.8 NC Bluetooth: Bluetooth PAN Encryption

7.2.3.4.8.1

[General] PAN Profile Secrecy Modes:

a) Clear Mode

b) Encrypted mode (at either the baseband or service level).

Recommendation of use could be reflected in the selection of device default values.

Supporting means that capability shall be implemented and available, but not necessarily used.

7.2.3.4.8.2

[Guideline] If Bluetooth is supported, then the PAN Profile v1.0 Secrecy Mode 2 (encrypted mode) on the Bluetooth baseband level shall be supported.

[Attributes]

M	R	n/a	n/a	n/a	[80] [82]	QLFGY	

7.2.3.4.8.3

[Guideline] If Bluetooth is supported, then the PAN Profile v1.0 Secrecy Mode 2 (encrypted mode) on the Bluetooth baseband level should be used when communicating.

[Attributes]

S	L	n/a	n/a	n/a	[82]	2RQFZ	

7.2.3.4.9 NC Bluetooth: BTH link key length requirement

[Guideline] If Bluetooth is supported, then the Bluetooth encryption key length shall between 8 and 16 octets.

[Attributes]

M	L	n/a	n/a	n/a	[82]	GV6YK	

NOTE: This is the Bluetooth baseband level encryption key.

7.2.3.4.10 NC Bluetooth: DLNAQOS Tolerance

[Guideline] If Bluetooth is supported, then incoming tagged packets shall be tolerated. Tagged packets are Ethernet packets encapsulated using BNEP that include priority tags conformant with [3], subclause 3.5, entitled 'Elements of the Tagged MAC Frame'. Here, 'tolerate' means that the packet payload of any received tagged or untagged packet shall be properly passed up to the higher layers in the network stack.

[Attributes]

M	R	n/a	n/a	n/a	[79] [3]	WPW7R	

NOTE: The BNEP Specification states that these tags need to be properly carried over BNEP. Note that Bluetooth devices are not required to enforce QoS over the Bluetooth link.

7.2.3.4.11 NC-PS Modes: Bluetooth Specific Mapping

7.2.3.4.11.1

[Guideline] If Bluetooth is supported, then the 'Active' NC-PS mode of the connection between a PANU and NAP shall only be mapped to the existence of a Bluetooth link in the 'Active' Bluetooth mode.

[Attributes]

M	A	n/a	n/a	n/a	[82] [83]	GZEEN	

NOTE: The NC-PS modes are bearer-independent definitions and are instantiated (mapped) differently, depending on the specific bearer used.

7.2.3.4.11.2

[GUIDELINE] If Bluetooth is supported, then the 'Standby' NC-PS mode of the connection between an PANU and NAP shall only be mapped to the existence of a Bluetooth link in the 'Sniff' Bluetooth low power mode.

[ATTRIBUTES]

M	A	n/a	n/a	n/a	[82] [83]	RMVW6	

NOTE: This guideline specifies how this mapping is done for Bluetooth.

7.2.3.4.11.3

[GUIDELINE] If Bluetooth is supported, then the 'Disconnected' NC-PS mode of the connection between a PANU and NAP shall only be mapped to the absence of a Bluetooth link between them.

[ATTRIBUTES]

M	A	n/a	n/a	n/a	[82] [83]	3QQ46	

7.2.3.4.11.4

[GUIDELINE] If Bluetooth is supported, then Bluetooth low power modes not mapped to any NC-PS mode, as specified in this guideline, should not be used for connectivity to the home network.

[ATTRIBUTES]

S	A	n/a	n/a	n/a	[82] [83]	XRIP7	

NOTE: This version of the guidelines does not include other Bluetooth low power modes (i.e. 'Hold' and 'Park') as part of the NC-PS modes scheme for Bluetooth. Their use by devices when connecting to the home network is discouraged, as it can lead to behavior non-compliant with some of these guidelines.

7.2.3.5 General Capability Requirements: Network Connectivity Power Savings (NC-PS)

7.2.3.5.1 NC-PS modes

7.2.3.5.1.1

[GUIDELINE] If the NC-PS bearer level power saving scheme is supported, then there shall be support for the 'Active' and 'Disconnected' NC-PS modes, as specified in these guidelines for the bearer used.

[ATTRIBUTES]

M	A	n/a	n/a	n/a	n/a	Q5NZM	

NOTE: Devices need to be able to map NC-PS modes to the corresponding bearer-specific modes, as specified in these guidelines.

The NC-PS modes are mapped differently depending on the specific bearer used.

7.2.3.5.1.2

[GUIDELINE] If the NC-PS bearer level power saving scheme is supported, then infrastructure devices that are used to connect other devices to the home network shall support the 'Standby' NC-PS mode, as specified in these guidelines for the bearer used.

[ATTRIBUTES]

M	A	n/a	n/a	n/a	n/a	8V64Y	

7.2.3.5.1.3

[GUIDELINE] If the NC-PS bearer level power saving scheme is supported, then devices that use infrastructure devices to connect to the home network should support the 'Standby' NC-PS mode, as specified in these guidelines for the bearer used.

[ATTRIBUTES]

S	A	n/a	n/a	n/a	n/a	46AZS	

7.2.3.5.1.4

[GUIDELINE] If the NC-PS bearer level power saving scheme is supported, then devices shall be able to report the current NC-PS mode information of the connection between them as one of the NC-PS modes defined in these guidelines: Active, Standby, Disconnected.

[ATTRIBUTES]

M	A	n/a	n/a	n/a	n/a	8VUO2	

NOTE: The Devices need to have a common understanding of the current NC-PS mode of the connection between them.

This allows entities, such as applications or UPnP-level agents/proxies, to know the current NC-PS mode.

7.2.3.5.2 NC-PS modes: Transition control

7.2.3.5.2.1

[GUIDELINE] If the NC-PS bearer level power saving scheme is supported, then transitions between the 'Active' and 'Standby' NC-PS modes shall be initiated only by the device that uses an infrastructure device to connect to the home network and not by that infrastructure device.

[ATTRIBUTES]

M	A	n/a	n/a	n/a	n/a	LFGY4	

NOTE: The mobile device, as the most power constrained end of the two, knows best when the connection needs to be in a power savings mode and the mobile device needs to be in control.

7.2.3.5.2.2

[GUIDELINE] If the NC-PS bearer level power saving scheme is supported, then both ends of a link shall be able to force transition to the 'Disconnected' NC-PS mode at any time, by tearing-down the link of the underlying bearer.

[ATTRIBUTES]

M	A	n/a	n/a	n/a	n/a	RQFZZ	

NOTE: Nothing can prevent the underlying bearer link to be broken (on purpose or accidentally) by either end of the connection. The 'Disconnected' NC-PS mode is entered implicitly upon disconnection.

7.2.3.5.2.3

[GUIDELINE] If the NC-PS bearer level power saving scheme is supported, then both ends of a communication link shall decide that the transition to the 'Disconnected' NC-PS mode is completed upon detecting the loss of the link of the underlying bearer.

[ATTRIBUTES]

M	A	n/a	n/a	n/a	n/a	V6YK2	

NOTE: There is no need for any request and reply for the transition to the 'Disconnected' NC-PS mode

7.2.3.5.2.4

[GUIDELINE] If the NC-PS bearer level power saving scheme is supported, then devices should be able to allow control of the permissible, under these guidelines, transitions between NC-PS modes.

[ATTRIBUTES]

S	A	n/a	n/a	n/a	n/a	PW7RV	

NOTE: This allows entities, such as applications or UPnP-level agents/proxies, to control the allowed transitions between NC-PS modes.

7.2.3.5.3 NC-PS Modes: Control of Bearer PS Parameters

7.2.3.5.3.1

[GENERAL] The type and range of parameters are different for each bearer. To ensure interoperability, minimum range requirements are specified in these guidelines for supported bearers.

7.2.3.5.3.2

[GUIDELINE] If the NC-PS bearer level power saving scheme is supported, then any relevant parameters of the power savings mechanism of the underlying bearer shall be requested by the device that uses an infrastructure device to connect to the home network, from that infrastructure device.

[ATTRIBUTES]

M	A	n/a	n/a	n/a	n/a	ZEENJ	

7.2.3.5.3.3

[GUIDELINE] If the NC-PS bearer level power saving scheme is supported, then infrastructure devices that are used to connect other devices to the home network shall respond positively to the parameters requested by these devices.

[ATTRIBUTES]

M	A	n/a	n/a	n/a	n/a	MVW6Y	

7.2.3.6 General Capability Requirements for MoCA

7.2.3.6.1 NC MoCA: Connector

[GUIDELINE] If MoCA is supported, then a 75 ohm Female F-Connector is required.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[58]	HWWTE	

7.2.3.6.2 NC MoCA: MoCA Conformance

[GUIDELINE] If MoCA is supported, the implementation shall conform to the MoCA Specification and MoCA Certification Test Plan requirements at the time the product is offered to the market.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[97] [98]	M6GKZ	

NOTE: MoCA is the Multimedia over Coax Alliance - an industry consortium that defines specifications for networking over the in-home coaxial cable [97].

7.2.3.7 General Capability Requirements for HPNA

7.2.3.7.1 NC HPNA: Connector

[GUIDELINE] If HPNA is supported, then a 75 ohm Female F-Connector is required.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[58]	STTML	N

7.2.3.7.2 NC HPNA: HPNA Conformance

[GUIDELINE] If HPNA is supported, the implementation shall conform to the HPNA Specification and HPNA Certification Test Plan requirements at the time the product is offered to the market.

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[59] [102]	V5HT4	N

7.2.4 Networking and Connectivity: QoS Requirements

7.2.4.1 General

The guidelines in this subclause provide requirements for priority-based QoS, hereinafter referred to as DLNAQOS. With DLNAQOS, applications label (tag) packets with the User Priority (UP) that dictates how the packets are allowed to access the network media and device queues. The DLNAQOS guidelines are contained in several subclauses as illustrated in Figure 26. Table 11 summarizes the default DLNAQOS (User Priority) tag correlation between specific DLNA traffic types and different network media types. In this table, streaming, interactive, and background transfers are as defined in Table 46. The default priorities, or lower, are used if DLNAQOS is implemented. It is not permitted to use priorities above the default stated.

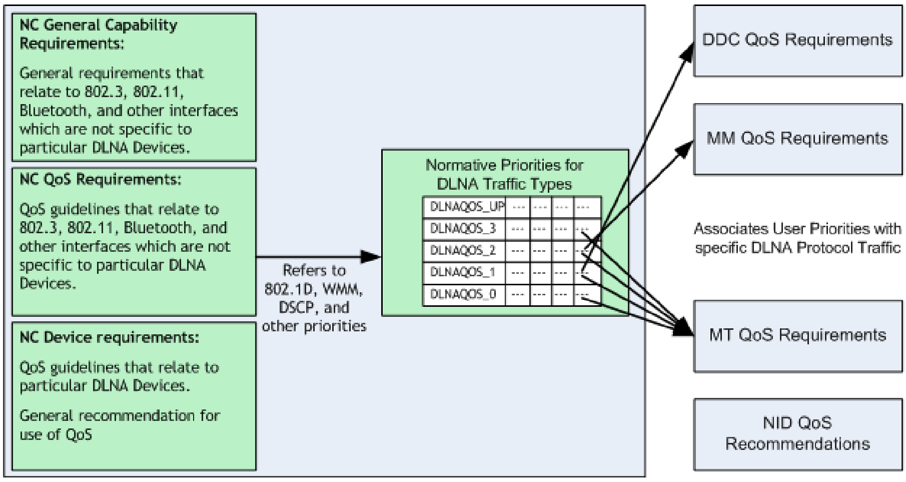


Figure 26 — DLNA QoS Visual Organization

While there are eight priority categories defined by 802.1D Annex G [1] and HPNA [59], Wi-Fi WMM [85], [86] only defines four categories and MoCA [97] only defines three. Table 11 directly correlates 802.1Q [2], WMM, MoCA, HPNA and DSCP [32] tags without any overlap, i.e. multiple 802.1Q/DSCP values for a single WMM access class or MoCA Priority.

Table 11 — Normative Priorities for DLNA Traffic Types

DLNAQOS_UP	DLNA Traffic Types	802.1Q User Priority	WMM Access Category	MoCA Priority	HPNA	DSCP	Subclause Details
DLNAQOS_3 (Highest)	• TCP messages generated by Content Receivers • DLNA Link Protection key exchange messages	7	VO	High	6	0x38	7.5.4.4.2.18 8.3.2 in [57] 9.2.2 in [57]
DLNAQOS_2	• Audio-only or A/V Streaming Transfers • UPnP AVTransport stream control • RTCP messages generated by Content Sources • RTSP messages	5	VI	Medium	5	0x28	7.5.4.2.12 7.4.1.6.22 7.5.4.4.2.18 7.5.4.4.6.2.81
DLNAQOS_1	• efault priority for any traffic defined by DLNA guidelines, unless specified otherwise • Interactive transfers • Remote User Interface messages	0	BE	Low	2	0x00	7.3.2.36 7.5.4.2.11 7.8.2.8

DLNAQOS_UP	DLNA Traffic Types	802.1Q User Priority	WMM Access Category	MoCA Priority	HPNA	DSCP	Subclause Details
DLNAQOS_0 (Lowest)	• Background transfers	1	BK	Low	0	0x08	7.5.4.2.10 7.8.2.8

7.2.4.2 DLNAQOS Requirements: Ethernet

7.2.4.2.1 NC Ethernet DLNAQOS: Conformance

[GUIDELINE] If DLNAQOS is supported on an Ethernet network interface, then it shall be compliant with all mandatory requirements for tagging where Tagged packets are Ethernet packets that include priority tags conformant with IEEE 802.3 [3], subclause 3.5, entitled 'Elements of the Tagged MAC Frame' and clause 9 of IEEE 802.1Q [2].

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[2] [3]	QQ463

NOTE: Packet tagging is the QoS mechanism used for wired networks.

7.2.4.2.2 NC Ethernet DLNAQOS: Tagging

7.2.4.2.2.1

[GUIDELINE] If DLNAQOS is supported on an Ethernet network interface, then the implementation shall apply both the 802.1Q VLAN priority tag (except as noted in 7.2.4.2.2.4) as well as the DSCP tag to outgoing traffic in accordance with the DLNAQOS_UP value in Table 11 or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3).

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[2] [3]	RIP7M

NOTE: It is not permitted to use priorities above the values specified in Table 11.

For exchanges involving a request and response, the implementation returning a response might not know the required DLNAQOS_UP value until it has parsed the request. It is at that time when it needs to apply the appropriate DLNAQOS_UP value. Note: There can be TCP network traffic during connection establishment that uses an inappropriate DLNAQOS_UP value.

For an HTTP streaming operation, the server needs to ensure the appropriate DLNAQOS_UP value (or lower) is used for the Entity Body.

7.2.4.2.2.2

[GUIDELINE] The phrase "or a lower DLNAQOS_UP" value means that the highest permitted DLNAQOS_UP value should be used.

[ATTRIBUTES]

S	A	n/a	n/a	n/a	n/a	5NZMS

7.2.4.2.2.3

[GUIDELINE] The phrase "or a lower DLNAQOS_UP" value also means that a lower DLNAQOS_UP value (indicated in the guideline) may be used.

[Attributes]

O	A	n/a	n/a	n/a	n/a	V64Y4	

7.2.4.2.2.4

[Guideline] For best-effort traffic on Ethernet, the implementation may omit the 802.1Q VLAN tag because frames with no tag are handled best-effort by default.

[Attributes]

O	R	n/a	n/a	n/a	[2] [3]	6AZSX	

7.2.4.3 DLNAQOS Requirements: 802.11

7.2.4.3.1 NC 802.11 DLNAQOS: Conformance

[Guideline] If DLNAQOS is supported on an 802.11 network interface, then it shall conform to all mandatory requirements in the WiFi WMM Test Plan [86] and specification [85].

[Attributes]

M	R	n/a	n/a	n/a	[85] [86]	VUO2Z	

NOTE: QoS support is optional, but if supported, it needs to conform to Wi-Fi requirements.

WMM provides the base level QoS specification for 802.11 network devices.

7.2.4.3.2 NC 802.11 DLNAQOS: Tagging

[Guideline] If DLNAQOS is supported on an 802.11 network interface, then the implementation shall apply both the WMM tag as well as the DSCP tag to outgoing traffic in accordance with the appropriate DLNAQOS_UP value in Table 11 or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3).

[Attributes]

M	R	n/a	n/a	n/a	[2] [3] [86] [32]	FGY4S	

NOTE: 802.1Q and DSCP correlation is in agreement with the WMM test plan recommendation.

It is not permitted to use priorities above the values specified in Table 11.

For exchanges involving a request and response, the implementation returning a response might not know the required DLNAQOS_UP value until it has parsed the request. It is at that time when it needs to apply the appropriate DLNAQOS_UP value. Note: There can be TCP network traffic during connection establishment that uses an inappropriate DLNAQOS_UP value.

7.2.4.4 DLNAQOS Requirements: Bluetooth

7.2.4.4.1 NC Bluetooth DLNAQOS: Conformance

[Guideline] If DLNAQOS is supported on a Bluetooth network interface, then it shall be compliant with all mandatory requirements for tagging Ethernet traffic which is encapsulated by BNEP, where Tagged packets are Ethernet packets that include priority tags conformant with IEEE 802.3 [3], subclause 3.5, entitled 'Elements of the Tagged MAC Frame' and clause 9 of IEEE 802.1Q [2].

[Attributes]

M	R	n/a	n/a	n/a	[79] [2] [3]	GY4SO	

NOTE: Packet tagging is the QoS mechanism used for wired networks. BNEP specifies how tagged Ethernet packets are encapsulated.

7.2.4.4.2 NC Bluetooth DLNAQOS: Tagging

7.2.4.4.2.1

[GUIDELINE] If DLNAQOS is supported on a Bluetooth network interface, then the implementation shall apply both the 802.1Q VLAN priority tag (except as noted in 7.2.4.4.2.2) as well as the DSCP tag to outgoing Ethernet traffic which is encapsulated by BNEP, in accordance with the DLNAQOS_UP value in Table 11 or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3).

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[79] [2] [3]	UO2ZU	

NOTE: It is not permitted to use priorities above the values specified in Table 11.

For exchanges involving a request and response, the implementation returning a response might not know the required DLNAQOS_UP value until it has parsed the request. It is at that time when it needs to apply the appropriate DLNAQOS_UP value. Note: There can be TCP network traffic during connection establishment that uses an inappropriate DLNAQOS_UP value.

For an HTTP streaming operation, the server needs to ensure the appropriate DLNAQOS_UP value (or lower) is used for the Entity Body.

7.2.4.4.2.2

[GUIDELINE] If DLNAQOS is supported on a Bluetooth network interface, for best-effort Ethernet traffic, which is encapsulated by BNEP, then the implementation may simply omit the 802.1Q VLAN tag because frames with no tag are handled best-effort by default.

[ATTRIBUTES]

O	R	n/a	n/a	n/a	[79] [2] [3]	AZSXK	

NOTE: Many legacy network devices do not handle 802.1Q tags well (don't expect a larger frame header), and react unfavorably. Not marking best-effort traffic allows for the greatest interoperability with these devices.

The DSCP value for the best effort traffic is 0 anyway.

802.1Q and DSCP correlation is in agreement with the WMM test plan recommendation.

7.2.4.5 DLNAQOS Requirements: MoCA

7.2.4.5.1 NC MoCA DLNAQOS: Conformance

[GUIDELINE] If DLNAQOS is supported on a MoCA network interface, then it shall be compliant with all mandatory requirements for Ethernet tagging where Tagged packets are Ethernet packets that include priority tags conformant with IEEE 802.3 [3], subclause 3.5, entitled 'Elements of the Tagged MAC Frame' and clause 9 of IEEE 802.1Q [2].

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[2] [97]	TOVH8	

NOTE: Packet tagging is the QoS mechanism used for wired networks. MoCA requires an Ethernet convergence layer so that Ethernet packets are transported transparently across the MoCA network.

7.2.4.5.2 NC MoCA DLNAQOS: Tagging

[GUIDELINE] If DLNAQOS is supported on an MoCA network interface, then the implementation shall apply both the 802.1Q VLAN priority tag (except as noted in 7.2.4.2.2.4) as well as the DSCP tag to outgoing traffic in accordance with the DLNAQOS_UP value in Table 11 or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3).

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[2] [97]	Z4L9V	

NOTE: It is not permitted to use priorities above the values specified in Table 11.

For exchanges involving a request and response, the implementation returning a response might not know the required DLNAQOS_UP value until it has parsed the request. It is at that time when it is expected to apply the appropriate DLNAQOS_UP value. Note: There can be TCP network traffic during connection establishment that uses an inappropriate DLNAQOS_UP value.

For an HTTP streaming operation, the server needs to ensure the appropriate DLNAQOS_UP value (or lower) is used for the Entity Body

7.2.4.6 DLNAQOS Requirements: HPNA

7.2.4.6.1 NC HPNA DLNAQOS: Conformance

[GUIDELINE] If DLNAQOS is supported on an HPNA network interface, then it shall be compliant with all mandatory requirements for Ethernet tagging where Tagged packets are Ethernet packets that include priority tags conformant with IEEE 802.3 [3], subclause 3.5, entitled 'Elements of the Tagged MAC Frame' and clause 9 of IEEE 802.1Q [2].

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[2] [59]	39U8A	N

NOTE: Packet tagging is the QoS mechanism used for wired networks. HPNA requires an Ethernet convergence layer so that Ethernet packets are transported transparently across the HPNA network.

7.2.4.6.2 NC HPNA DLNAQOS: Tagging

[GUIDELINE] If DLNAQOS is supported on an HPNA network interface, then the implementation shall apply both the 802.1Q VLAN priority tag (except as noted in 7.2.4.2.2.4) as well as the DSCP tag to outgoing traffic in accordance with the DLNAQOS_UP value in Table 11 or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3).

[ATTRIBUTES]

M	R	n/a	n/a	n/a	[2] [59]	XQ6OB	N

NOTE: It is not permitted to use priorities above the values specified in Table 11.

For exchanges involving a request and response, the implementation returning a response might not know the required DLNAQOS_UP value until it has parsed the request. It is at that time when it is expected to apply the appropriate DLNAQOS_UP value. Note: There can be TCP network traffic during connection establishment that uses an inappropriate DLNAQOS_UP value.

For an HTTP streaming operation, the server needs to ensure the appropriate DLNAQOS_UP value (or lower) is used for the Entity Body

7.2.5 Networking and Connectivity: Device Requirements

7.2.5.1 General

The guidelines in this subclause specify the capabilities or combination of capabilities that DLNA devices specifically support. These specific device requirements reference the guidelines defined in 7.2.3 Networking and Connectivity: General Capability Requirements, Table 11, and 7.2.4 Networking and Connectivity: QoS Requirements. For example, a specific requirement for DLNA HND devices is that they shall support either Ethernet or 802.11 connectivity, where Ethernet and 802.11 capabilities are described in 7.2.3 Networking and Connectivity: General Capability Requirements and are referenced by name. Correspondingly, a specific requirement for DLNA MHD devices is that they shall support either Ethernet, 802.11, or Bluetooth connectivity, where Ethernet, 802.11 and Bluetooth capabilities are described in 7.2.3 Networking and Connectivity: General Capability Requirements and are referenced by name.

7.2.5.2 Device Requirements: Common

7.2.5.2.1 NC Devices: IP Stack

[GUIDELINE] DLNA Device Classes shall support a TCP/IP stack that includes IPv4, TCP, UDP, ARP, and ICMP components conformant to all required client aspects of [16].

[ATTRIBUTES]

M	R	HND	MHD	MIU	[11] [12] [13] [14] [15] [16]	64Y4F

NOTE: A DNS client is omitted because it is not strictly needed for UPnP operations on the network. Native IP addresses actually simplify the use of UPnP.

7.2.5.2.2 NC Devices: IP Address Acquisition

7.2.5.2.2.1

[GUIDELINE] DLNA Device Classes shall support DHCP client functionality [22] and obtain an IP address and subnet mask from a home network DHCP server if present. They shall implement Auto IP as defined by the UPnP Device Architecture v1.0 specification ([47]) so that if a DHCP server is not present on the home network, a link-local network address can be automatically acquired.

[ATTRIBUTES]

M	R	HND	MHD	MIU	[22] [47]	NZMSY

NOTE: This guideline includes an assumption that devices need to transition from DHCP to Auto-IP, when devices fail to renew an expired IP address (that was assigned by a DHCP server). Likewise, Auto-IP requires the devices to make periodic attempts (once every 5 minutes) to acquire a DHCP-assigned IP address, when operating with a self-assigned IP address.

Although it is desirable for a device using a DHCP-assigned IP address to exhibit interoperability with a device using an Auto-IP address, DLNA makes no requirement that full interoperability will occur in such scenarios.

7.2.5.2.2.2

[GUIDELINE] DLNA Device Classes that allocate addresses with the Auto-IP link local address allocation system of [47] should adhere to all aspects of subclause 2.7 of [7]. No packet generated by a DLNA Device Class with an Auto-IP allocated address as the source or destination address should be sent to a router for forwarding.

[ATTRIBUTES]

S	R	HND	MHD	MIU	[7] [47]	IP7MK

NOTE: Subclause 2.7 of [7] contains information how Auto-IP addresses are impacted by routed subnets.

7.2.5.2.2.3

[GUIDELINE] DLNA Device Classes that allocate addresses with the DHCP address allocation system should adhere to all aspects of subclause 2.7 of [7]. No packet generated by a DLNA Device Class with an Auto-IP allocated address as the destination address should be sent to a router for forwarding.

[ATTRIBUTES]

S	R	HND	MHD	MIU	[7]	Q463H

NOTE: Subclause 2.7 of [7] contains information how Auto-IP addresses are impacted by routed subnets.

7.2.5.2.2.4

[Guideline] DLNA Device Classes that allocate addresses with the Auto-IP link local address allocation system of [47] should attempt to interoperate with DLNA Device Classes that have allocated DHCP addresses as per clause 3 of [7] as to the interaction between link-local and non-link-local addresses. A DLNA Device Class should attempt to send all packets on the local link.

[Attributes]

S	R	HND	MHD	MIU	[7] [47]	VW6Y8

7.2.5.2.2.5

[Guideline] DLNA Device Classes that allocate addresses with the DHCP address allocation system should attempt to interact with endpoints that have allocated link-local Auto-IP addresses as per clause 3 of [7]. A DLNA Device Class should attempt to send all packets bound to a link-local address on the local link.

[Attributes]

S	R	HND	MHD	MIU	[7]	EENJB

7.2.5.2.2.6

[Guideline] When the lease on an IP address expires, and the DLNA Device Class is unable to renew the lease on that IP address, or obtain a lease on a new IP address, the Device Class shall use an Auto-IP address. The Auto-IP address can be acquired before or after the DHCP lease expires.

[Attributes]

M	A	HND	MHD	MIU	[7] [22] [47]	W7RVO

NOTE: A DLNA Device Class will not utilize an expired DHCP IP address. The Auto-IP address can be previously obtained and defended while the DHCP lease was active. Alternatively the Device Class can obtain a new Auto-IP address as defined in 7.2.5.2.2.1

7.2.5.2.3 NC Devices: DLNAQOS Support

7.2.5.2.3.1

[Guideline] DLNA Device Classes should support DLNAQOS on all network interfaces.

[Attributes]

S	A	HND	MHD	MIU	n/a	6YK2S

7.2.5.2.3.2

[Guideline] If DLNAQOS is supported on an Ethernet network interface by a DLNA Device Class, then it shall be conformant to all [NC Ethernet DLNAQOS:] labeled requirements in 7.2.4 Networking and Connectivity: QoS Requirements.

[Attributes]

M	A	HND	MHD	MIU	n/a	QFZZ7

7.2.5.2.3.3

[GUIDELINE] If DLNAQOS is supported on an 802.11 network interface by a DLNA Device Class, then it shall be conformant to all [NC 802.11 DLNAQOS:] labeled requirements in 7.2.4 Networking and Connectivity: QoS Requirements.

[ATTRIBUTES]

M	A	HND	MHD	MIU	n/a	FZZ7O	

7.2.5.2.3.4

[GUIDELINE] If DLNAQOS is supported on a Bluetooth network interface by a DLNA Device Class, then it shall be conformant to all [NC Bluetooth DLNAQOS:] labeled requirements in 7.2.4 Networking and Connectivity: QoS Requirements.

[ATTRIBUTES]

M	A	MHD	n/a	MIU	n/a	YK2SS	

7.2.5.2.3.5

[GUIDELINE] If DLNAQOS is supported on a MoCA network interface by a DLNA Device Class, then it shall be conformant to all [NC MoCA DLNAQOS:] labeled requirements in 7.2.4 Networking and Connectivity: QoS Requirements.

[ATTRIBUTES]

M	A	HND	MHD	MIU	n/a	6UAFT	

7.2.5.2.3.6

[GUIDELINE] If DLNAQOS is supported on an HPNA network interface by a DLNA Device Class, then it shall be conformant to all [NC HPNA DLNAQOS:] labeled requirements in 7.2.4 Networking and Connectivity: QoS Requirements.

[ATTRIBUTES]

M	A	HND	MHD	MIU	n/a	7L9NB	N

7.2.5.3 Device Requirements: HND

7.2.5.3.1 NC HND Devices: Required/Optional Connectivity

7.2.5.3.1.1

[GUIDELINE] DLNA Device Classes shall support at least one of the following connectivity selections:

- Ethernet conformant to all [NC Ethernet:] labeled requirements in the General Capability Requirements subclause of 7.2.3 Networking and Connectivity: General Capability Requirements.
- 802.11 conformant to all [NC 802:11:] labeled requirements in the General Capability Requirements subclause of 7.2.3 Networking and Connectivity: General Capability Requirements.

[ATTRIBUTES]

M	R	HND	n/a	MIU	n/a	7RVON	

7.2.5.3.1.2

[GUIDELINE] DLNA Device Classes may support the following connectivity selections:

- MoCA conformant to all [NC MoCA:] labeled requirements in the General Capability Requirements subclause of 7.2.3 Networking and Connectivity: General Capability Requirements.
- HPNA conformant to all [NC HPNA:] labeled requirements in the General Capability Requirements subclause of 7.2.3 Networking and Connectivity: General Capability Requirements.

[ATTRIBUTES]

O	R	HND	n/a	MIU	n/a	CD695	C

NOTE: MoCA and HPNA are optional networks and connectivity for DLNA Device Classes in the HND Device Category.

7.2.5.3.2 NC HND Devices: Recommended Connectivity

[GUIDELINE] DLNA Device Classes should support all of the following connectivity selections:

- Ethernet conformant to all [NC Ethernet:] labeled requirements in the General Capability Requirements subclause of 7.2.3 Networking and Connectivity: General Capability Requirements.
- 802.11 conformant to all [NC 802:11:] labeled requirements in the General Capability Requirements subclause of 7.2.3 Networking and Connectivity: General Capability Requirements.

Any of the above selections can be supported via an add on card, dongle, or equivalent..

[ATTRIBUTES]

S	R	HND	n/a	MIU	n/a	ZSXKY	

NOTE: This guideline is intended to ensure that a consumer does not have to understand the different network connectivity types when purchasing a DLNA product. A consumer will be assured a newly purchased product will work with other previously purchased DLNA products.

7.2.5.4 Device Requirements: MHD

7.2.5.4.1 NC MHD Devices: Required/Optional Connectivity

7.2.5.4.1.1

[GUIDELINE] DLNA Device Classes shall support at least one of the following connectivity selections:

- Ethernet conformant to all [NC Ethernet:] labeled requirements in the General Capability Requirements subclause of 7.2.3 Networking and Connectivity: General Capability Requirements
- 802.11 conformant to all [NC 802:11:] labeled requirements in the General Capability Requirements subclause of 7.2.3 Networking and Connectivity: General Capability Requirements.
- Bluetooth conformant to all [NC Bluetooth:] labeled requirements in the General capability requirements subclause of 7.2.3 Networking and Connectivity: General Capability Requirements.

[ATTRIBUTES]

M	R	MHD	n/a	MIU	n/a	O2ZUW	

7.2.5.4.1.2

[GUIDELINE] DLNA Device Classes may support the following connectivity selections:

- MoCA conformant to all [NC MoCA:] labeled requirements in the General Capability Requirements subclause of 7.2.3 Networking and Connectivity: General Capability Requirements.
- HPNA conformant to all [NC HPNA:] labeled requirements in the General Capability Requirements subclause of 7.2.3 Networking and Connectivity: General Capability Requirements.

[ATTRIBUTES]

O	R	MHD	n/a	MIU	n/a	F855B	C

NOTE: MoCA and HPNA are optional networks and connectivity for DLNA Device Classes in the HND Device Category.

7.2.5.4.2 NC MHD Devices: Connection to the Home Network

[GUIDELINE] If an M-NCF Device Class is detected for the desired bearer, a DLNA Device Class should use it to connect to the home network.

[ATTRIBUTES]

S	A	MHD	n/a	MIU	n/a	Y4SOW	

NOTE: It is up to the MHD (e.g. based on user or application preferences) to select the desired bearer to connect to the home network.

Connectivity for the desired bearer via an M-NCF, when possible, provides a better user experience. This recommendation could be reflected, for example, in the MHD's default settings.

Direct connectivity for the desired bearer can also occur via standard networking elements such as Access Points, switches or hubs, etc.

7.2.5.4.3 NC MHD Devices: Topology restriction

7.2.5.4.3.1

[GUIDELINE] A DLNA Device Class shall not be connected to more than one M-NCF at a time.

[ATTRIBUTES]

M	A	MHD	n/a	MIU	n/a	2ZUWV	

NOTE: To avoid using the spanning tree protocol in M-NCF.

7.2.5.4.4 NC MHD Devices: Bluetooth Device Discovery

7.2.5.4.4.1

[GUIDELINE] If Bluetooth is supported, then a DLNA Device Class shall be able to perform Bluetooth inquiry.

[ATTRIBUTES]

M	R	MHD	n/a	MIU	[82]	4SOW3	

NOTE: MHD enters the inquiry substate to perform Bluetooth device discovery.

7.2.5.4.4.2

[GUIDELINE] If Bluetooth is supported, then a DLNA device Class should filter inquiry responses based on the networking bit in the Class of Device field to narrow down the search for M-NCFs.

[ATTRIBUTES]

S	L	MHD	n/a	MIU	[81] [82]	SXKY8	

NOTE: This will reduce Bluetooth device discovery time.

7.2.5.4.5 NC MHD Devices: Bluetooth SDP

7.2.5.4.5.1

[GUIDELINE] If Bluetooth is supported, then a DLNA Device Class shall be able to retrieve the SDP record from an NAP.

[ATTRIBUTES]

M	R	MHD	n/a	MIU	[81] [82]	MSYVI	

NOTE: M-NCF is a NAP.

7.2.5.4.5.2

[GUIDELINE] If Bluetooth is supported, then a DLNA Device Class should not expose the 'M-NCF_Description' text portion of the ServiceDescription attribute specified in 7.2.5.5.15.4, in order to provide a better user experience.

[ATTRIBUTES]

S	A	MHD	n/a	MIU	[81] [82]	7MKJE	

NOTE: M-NCF advertises its specific DLNA capabilities that can be shown as BT device description. This specific description part is not in user friendly form and is therefore expected not to be shown.

7.2.5.4.6 NC MHD Devices: MHD Bluetooth PIN

[GUIDELINE] If Bluetooth is supported, then a DLNA Device Class shall implement support for variable PIN codes.

[ATTRIBUTES]

M	L	MHD	n/a	MIU	[82]	63HNQ	

NOTE: The use of fixed PIN in both the MHD device and M-NCF would preclude interoperability.

7.2.5.4.7 NC MHD Devices: BT PAN role

[GUIDELINE] If Bluetooth is supported, then an MHD Device Class shall perform the role of a PANU.

[ATTRIBUTES]

M	L	MHD	n/a	MIU	[81]	6Y8ZI	

7.2.5.4.8 NC MHD Devices: MHD device PAN compressed packet type

[GUIDELINE] If Bluetooth is supported, then BNEP Ethernet header compression (BNEP_COMPRESSED_ETHERNET_DEST_ONLY Packet type) should be used.

[ATTRIBUTES]

S	L	MHD	n/a	MIU	[81]	NJBS9	

NOTE: Reduces amount of redundant header information sent over point to point link.

7.2.5.4.9 NC MHD Devices: BT PS Mode support

[GUIDELINE] If Bluetooth is supported, then the 'Sniff' Bluetooth power savings mode should be supported.

[ATTRIBUTES]

S	R	MHD	n/a	MIU	[82]	RVON4	

7.2.5.4.10 NC MHD Devices: NC-PS Mode Support Using Bluetooth

7.2.5.4.10.1

[GUIDELINE] If Bluetooth is supported, then a DLNA Device Class shall support the NC-PS bearer level power saving scheme conformant to all [NC-PS modes:] labeled requirements in the General capability requirements subclause of 7.2.3 Networking and Connectivity: General Capability Requirements.

[ATTRIBUTES]

M	A	MHD	n/a	MIU	n/a	K2SS4	

NOTE: MHD devices use infrastructure devices to connect to the home network and and as such they are expected to support the 'Standby' NC-PS mode, as specified in 7.2.3.5.1.3.

7.2.5.4.10.2

[GUIDELINE] If Bluetooth is supported, then a DLNA Device Class should support the 'Standby' NC-PS mode.

[ATTRIBUTES]

S	A	MHD	n/a	MIU	n/a	ENJBS	

7.2.5.4.10.3

[GUIDELINE] If Bluetooth is supported, then a DLNA Device Class shall be the one to initiate the transition from 'Active' to 'Standby' NC-PS mode, by requesting to move the Bluetooth link from the 'Active' to the 'Sniff' Bluetooth mode, following the 'Sniff' parameter restrictions specified in 7.2.5.4.11.

[ATTRIBUTES]

M	A	MHD	n/a	MIU	[82]	W6Y8Z	

7.2.5.4.10.4

[GUIDELINE] If Bluetooth is supported, then a DLNA Device Class shall be the one to request at any time the transition from 'Standby' to 'Active' NC-PS mode, by sending an LMP request to move the link from 'Sniff' to 'Active' Bluetooth mode.

[ATTRIBUTES]

M	A	MHD	n/a	MIU	[82]	463HN	

7.2.5.4.10.5

[GUIDELINE] If Bluetooth is supported, while the connection is in 'standby' NC-PS mode, then a change in parameters of the 'sniff Bluetooth power savings mode´can be requested at any time, following the 'Sniff' parameter restrictions specified in 7.2.5.4.11

[ATTRIBUTES]

O	A	MHD	n/a	MIU	[82]	P7MKJ	

NOTE: Requesting a change of the parameters of the current 'Sniff' Bluetooth power savings mode does not trigger a transition out of the 'Standby' mode.

7.2.5.4.11 NC MHD Devices: requested 'Sniff' parameter restrictions

7.2.5.4.11.1

[GUIDELINE] If Bluetooth is supported, then all sniff parameter requests sent from a DLNA Device Class to the M-NCF shall meet the restrictions outlined in this guideline.

[ATTRIBUTES]

M	A	MHD	n/a	MIU	[82]	ZMSYV	

7.2.5.4.11.2

[GUIDELINE] If Bluetooth is supported and a DLNA Device Class determines from the 'M-NCF_Description' text specified in 7.2.5.5.15.4 that the M-NCF supports ARP proxying, then any requested HCI Sniff_Max_Interval value shall be less than or equal to 20 seconds.

[ATTRIBUTES]

M	A	MHD	n/a	MIU	[82]	4Y4FH	

7.2.5.4.11.3

[GUIDELINE] If Bluetooth is supported and if a DLNA Device Class determines from the 'M-NCF_Description' text specified 7.2.5.5.15.4 in that the M-NCF does not support "ARP proxying", then any requested HCI Sniff_Max_Interval value shall be less than or equal to 2 seconds.

[ATTRIBUTES]

M	A	MHD	n/a	MIU	[81] [82]	ZZ7OR	

7.2.5.4.11.4

[GUIDELINE] If Bluetooth is supported, any HCI 'Sniff_Min_Interval' request shall be less or equal to the HCI Sniff_Max_Interval.

[ATTRIBUTES]

M	R	MHD	n/a	MIU	[82]	Z7ORX	

7.2.5.4.11.5

[GUIDELINE] If Bluetooth is supported and if a DLNA Device Class determines from the 'M-NCF_Description' text specified in 7.2.5.5.15.4 that the M-NCF supports "ARP proxying", then any HCI 'Sniff_Attempt' request should be greater or equal to 625 milliseconds.

[ATTRIBUTES]

S	A	MHD	n/a	MIU	[81] [82]	2SS47	

NOTE: This will ensure a minimum "duty cycle" while the BTH link between MHD and NCF is in the 'Sniff' mode and, therefore, a minimum maintained Bandwidth.

7.2.5.4.11.6

[GUIDELINE] If Bluetooth is supported and if a DLNA Device Class determines from the 'M-NCF_Description' text specified in 7.2.5.5.15.4 that the M-NCF does not support "ARP proxying", then any HCI 'Sniff_Attempt' request should be greater or equal to 62.5 milliseconds.

[ATTRIBUTES]

S	A	MHD	n/a	MIU	[82]	VON4B	

7.2.5.4.11.7

[GUIDELINE] If Bluetooth is supported, any HCI Sniff_Timeout request shall be greater than or equal to the HCI Sniff_Max_Interval.

[ATTRIBUTES]

M	R	MHD	n/a	MIU	[82]	JBS9Y	

7.2.5.4.12 NC MHD Devices: Bluetooth Establishing Network Access Rights

7.2.5.4.12.1

[GUIDELINE] If Bluetooth is supported a DLNA Device Class shall be able to initiate a request for network access rights from an M-NCF by initiating a pairing procedure with the M-NCF, as specified in 7.2.3.4.6

[ATTRIBUTES]

M	R	MHD	n/a	MIU	[82]	Y8ZIV	

NOTE: Network access right is the pre-approval for future network connection requests from an MHD. Connection requests from an MHD with permanent network access rights are always granted. Connection requests from an MHD with temporary network access rights are granted if other additional conditions are satisfied.

Network access rights need to be established individually because each M-NCF has its own Bluetooth Device Database.

Pairing can be initiated either by the MHD or the M-NCF.

7.2.5.4.12.2

[GUIDELINE] If Bluetooth is supported, any DLNA Device Class may initiate the Bluetooth pairing process to establish network access rights.

[ATTRIBUTES]

O	R	MHD	n/a	MIU	[82]	3HNQY	

7.2.5.4.12.3

[GUIDELINE] If Bluetooth is supported and multiple M-NCFs are in range, then a DLNA Device Class shall establish network access rights with each M-NCF individually.

[ATTRIBUTES]

M	A	MHD	n/a	MIU	[82]	MKJE8	

7.2.5.4.13 NC MHD Devices: Link Key Usage and Management

7.2.5.4.13.1

[GUIDELINE] If Bluetooth is supported and if there is a record of the DLNA Device Class in the M-NCF's Bluetooth Device Database, then the DLNA Device Class and the M-NCF shall use the link key stored in the database for authentication and deriving the encryption key..

[ATTRIBUTES]

M	A	MHD	n/a	MIU	[79] [82]	SYVI7	

7.2.5.5 Device Requirements: M-NCF

7.2.5.5.1 NC M-NCF: Required Connectivity

[GUIDELINE] An M-NCFshall have network interfaces to connect to the following two connectivity domains:

- The home connectivity domain with the bearers specified in 7.2.5.3.1.
- The mobile connectivity domain with Bluetooth conformant to all [NC Bluetooth:] labeled requirements in the General capability requirements subclause of 7.2.3 Networking and Connectivity: General Capability Requirements.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	n/a	4FHOQ	

NOTE: The M-NCF bridges the connectivity gap between the HND and MHD connectivity domains.

The current version of guidelines specifies only a 'BT M-NCF'.

7.2.5.5.2 NC M-NCF: Bridging

7.2.5.5.2.1

[GUIDELINE] An M-NCF shall perform layer-two bridging between home connectivity domain bearer interfaces and the mobile connectivity domain bearer interfaces according to 802.1D, as further specified by the PAN profile [81].

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	[81] [1]	XKY8O	

NOTE: Filtering of packets at the M-NCF can be performed for power saving.

7.2.5.5.2.2

[GUIDELINE] An M-NCF shall not do any additional filtering other than what is specified in [1] and 7.2.5.5.9.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	[81] [1]	ZUWVW	

7.2.5.5.3 NC M-NCF: IP Protocol Stack

[GUIDELINE] An M-NCF should support a TCP/IP stack that includes IPv4, TCP, UDP, ARP, and ICMP components conformant to all required client aspects of [16].

[ATTRIBUTES]

S	R	n/a	n/a	M-NCF	[11] [12] [13] [14] [15] [16]	SOW3V	

NOTE: TCP/IP stack support is not mandatory for M-NCF operations, but it can be used for proving device management e.g. though SNMP or web-based interface

A DNS client is omitted because it is not strictly needed for UPnP operations on the network. Native IP addresses actually simplify the use of UPnP.

7.2.5.5.4 NC M-NCF: IP Address Acquisition

[GUIDELINE] An M-NCF should support address allocation equivalent to all requirements on MHD and HND devices as specified in 7.2.5.2.2

[ATTRIBUTES]

S	R	n/a	n/a	M-NCF	[7] [22] [47]	OW3V2	

7.2.5.5.5 NC M-NCF: Multiple device support

[GUIDELINE] An M-NCF shall allow connections to multiple active client devices simultaneously.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	n/a	KY8OP	

7.2.5.5.6 NC M-NCF: ARP proxying functionality

7.2.5.5.6.1

[GUIDELINE] An M-NCF should support ARP proxying functionality as specified in the PAN profile [81] and further clarified in this guideline.

[ATTRIBUTES]

S	C	n/a	n/a	M-NCF	[81]	FHOQR	

7.2.5.5.6.2

[GUIDELINE] If an M-NCF supports the ARP proxying functionality, then it shall do so as follows:

- It shall first obtain the IP and MAC address of an MHD after the MHD is connected to it.
- After obtaining the MHD's IP and MAC address, it shall stop forwarding any ARP requests to the MHD and it shall reply to any ARP requests targeted at the MHD's IP address using the MHD's MAC address.

[ATTRIBUTES]

M	C	n/a	n/a	M-NCF	[81]	YVI7W	

7.2.5.5.7 NC M-NCF: NC-PS Conformance

7.2.5.5.7.1

[GUIDELINE] An M-NCF shall support the NC-PS bearer level power saving scheme conformant to all [NC-PS modes:] labeled requirements in the General capability requirements subclause of table 7.2.3 Networking and Connectivity: General Capability Requirements.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	n/a	KJE8I	

NOTE: The M-NCF is an infrastructure device that is used to connect other devices to the home network and as such it needs to support the NC-PS mode as specified in 7.2.3.5.1.2.

7.2.5.5.7.2

[GUIDELINE] An M-NCF shall support the 'Standby' NC-PS mode.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	n/a	HNQYX	

NOTE: An M-NCF needs to support the 'Standby' NC-PS mode

7.2.5.5.8 NC M-NCF: NC-PS Connection Individuality

7.2.5.5.8.1

[GUIDELINE] An M-NCF shall allow each connection to an MHD to be in a separate NC-PS mode, independently of the NC-PS mode of other connections.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	n/a	8ZIVT	

7.2.5.5.9 NC M-NCF: NC-PS Traffic Reduction Operations

7.2.5.5.9.1

[GUIDELINE] While the connection between the DLNA Device Class and the MHD is in the 'Active' NC-PS mode, an M-NCF shall not perform any traffic reduction operations that might reduce functionality at the IP-layer and above as experienced by the MHD device.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	n/a	BS9YU	

NOTE: Examples of operations that would reduce functionality of the MHD at the IP-layer and above are filtering of any DHCP-related traffic and filtering of any UPnP-related traffic.

7.2.5.5.9.2

[GUIDELINE] While the connection between the DLNA Device Class and the MHD is in the 'Standby' mode, an M-NCF should implement traffic reduction operations to prevent any or all of the following traffic from being received by the MHD:

- UPnP multicast traffic directed to address 239.255.255.250:1900
- ARP messages, provided that the DLNA Device Class also performs the ARP proxying functionality specified in 7.2.5.5.6 on behalf of the MHD device.

[ATTRIBUTES]

S	A	n/a	n/a	M-NCF	[15] [47]	ON4BV	

NOTE: Reducing the traffic flowing in the link between the MHD and M-NCF, results in power savings for the MHD device since the two ends need to spend less time communicating.

In 'Standby' mode an MHD device can receive (also send) unicast IP traffic, but might not receive UPnP multicast traffic if the M-NCF filters it. As a result, the MHD's full network state is maintained but its UPnP state can be lost.

7.2.5.5.10 NC M-NCF: Bluetooth Discovery

7.2.5.5.10.1

[GUIDELINE] If Bluetooth is supported, then an M-NCF should be in Bluetooth discoverable mode.

[ATTRIBUTES]

S	L	n/a	n/a	M-NCF	[82]	SS47L	

NOTE: The M-NCF is expected to perform Bluetooth Inquiry Scan to allow for Bluetooth discovery.

7.2.5.5.10.2

[GUIDELINE] If Bluetooth is supported, then an M-NCF shall set the Networking bit of the Class of Device (CoD) field.

[ATTRIBUTES]

M	R	n/a	n/a	M-NCF	[82]	7ORX3	

NOTE: Setting the CoD accelerates M-NCF Bluetooth discovery by inquiring MHDs.

7.2.5.5.11 NC M-NCF: Bluetooth Connectability

[GUIDELINE] If Bluetooth is supported, then an M-NCF shall be in Bluetooth connectable mode.

[ATTRIBUTES]

M	L	n/a	n/a	M-NCF	[82]	S47LY	

NOTE: The M-NCF is expected to perform Bluetooth Page Scan to allow for incoming connections.

7.2.5.5.12 NC M-NCF: Bluetooth compressed packet type

[GUIDELINE] If Bluetooth is supported, an M-NCF should use the BNEP Ethernet header compression (BNEP_COMPRESSED_ETHERNET_SOURCE_ONLY Packet type).

[ATTRIBUTES]

S	L	n/a	n/a	M-NCF	[81]	N4BVT	

NOTE: Reduces amount of redundant header information sent over point to point link.

7.2.5.5.13 NC M-NCF: Bluetooth PAN role

[GUIDELINE] If Bluetooth is supported, then an M-NCF shall perform the role of a NAP, as specified in the PAN profile.

[ATTRIBUTES]

M	L	n/a	n/a	M-NCF	[81]	S9YUG	

7.2.5.5.14 NC M-NCF: Bluetooth PAN mode

[GUIDELINE] If Bluetooth is supported, then an M-NCF shall operate in the NAP multi-user mode as specified in the PAN profile.

[ATTRIBUTES]

M	L	n/a	n/a	M-NCF	[81]	ZIVT6	

NOTE: This implies that a master-slave role switch upon connection might be necessary, so that the M-NCF is always the master in all Bluetooth connections.

7.2.5.5.15 NC M-NCF: Bluetooth SDP

7.2.5.5.15.1

[GUIDELINE] If Bluetooth is supported, then an M-NCF shall act as Bluetooth SDP server.

[ATTRIBUTES]

M	L	n/a	n/a	M-NCF	[81] [82]	NQYXG	

7.2.5.5.15.2

[GUIDELINE] If Bluetooth is supported, then an M-NCF shall create a NAP service record in the service discovery database as specified in [81].

[ATTRIBUTES]

M	R	n/a	n/a	M-NCF	[81]	JE8IW	

7.2.5.5.15.3

[GUIDELINE] If Bluetooth is supported, then the NAP service record of an M-NCF shall use one of the following values for the NetAccessType attribute:

- 0x0004 if the home connectivity domain bearer is 10Mb Ethernet
- 0x0005 if the home connectivity domain bearer is 100Mb Ethernet
- 0xFFFE if the home connectivity domain bearer is Ethernet at other rates, or if the HNv1 bearer is 802.11.

[ATTRIBUTES]

M	L	n/a	n/a	M-NCF	[81]	HOQRJ	

NOTE: The PAN profile has not assigned attribute values to represent 802.11 or Ethernet at rates other than 10Mb and 100Mb as the NetAccessType.

7.2.5.5.15.4

[GUIDELINE] If Bluetooth is supported, then the NAP SDP record (see [82] Part E) of an M-NCF shall include in the ServiceDescription attribute value the 'M-NCF_Description' text of the form "DLNAv1.5_MNCF_XXX_YYY", where the text options XXX and YYY are specified below, to indicate that it is a DLNA v1.5 M-NCF and its supported functionality:

- (a) XXX is equal to "FOF", if the DLNA Device Class does not support the UPnP multicast filtering functionality specified in guideline 7.2.5.5.9.2.
- (b) XXX is equal to "FON", if the DLNA Device Class supports the UPnP multicast filtering functionality specified in guideline 7.2.5.5.9.2 and will always apply filtering when the connection between an MHD and the M-NCF is in the 'Standby' NC-PS mode.
- (c) YYY is equal to "POF", if the DLNA Device Class does not support the ARP proxying functionality specified in guideline 7.2.5.5.6.
- (d) YYY is equal to "PON", if the DLNA Device Class supports the ARP proxy functionality specified in guideline7.2.5.5.6 and will always perform the ARP proxying functionality when the connection between the MHD and the DLNA Device Class is in the 'Standby' NC-PS mode.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	[82]	LPOHR	

NOTE: The ServiceDescription attribute is a human-readable text containing a brief description of the Bluetooth service. The MHD will be able to tell whether a Bluetooth NAP is a DLNA compliant Bluetooth M-NCF by looking at the ServiceDescription attribute value of the NAP SDP record. The MHD will also know whether the M-NCF is filtering UPnP multicast messages or whether it is performing ARP proxying on its behalf.

7.2.5.5.15.5

[GUIDELINE] If Bluetooth is supported, then an M-NCF shall prevent the 'M-NCF_Description' text portion of the ServiceDescription attribute value from being edited or deleted.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	[82]	Y8OP4

NOTE: Only the M-NCF will have control of editing the 'M-NCF_Description' text portion of the ServiceDescription field. It needs to not allow other entities, e.g. users or other applications from overwriting this portion of the field.

7.2.5.5.16 NC M-NCF: Bluetooth Device Database Definition

7.2.5.5.16.1

[GUIDELINE] If Bluetooth is supported, then an M-NCF device shall maintain, or have full access to, a Bluetooth Device Database to store security-related information on devices. The Bluetooth device database shall be stored in non-volatile memory.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	[80]	W3V2G

NOTE: Bluetooth device database is a non-volatile storage and need not be co-located with the M-NCF. The term database is used as defined in [80] and does not refer to specific implementations, but only to a general information storage facility.

7.2.5.5.16.2

[GUIDELINE] If Bluetooth is supported, then the Bluetooth Device Database of an M-NCF device shall, at a minimum, store the following information about an MHD device:

- BD_ADDR
- Device name
- Link key
- Authorized for PAN service
- Trusted or untrusted

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	[80] [81] [82]	3V2GQ

NOTE: "Authorized for PAN service" indicates whether future PAN connection requests from an untrusted device are automatically authorized.

Note: Trust Level field in device database differentiates between trusted and untrusted devices, which have fixed relationship (paired and have link key).

Unknown devices, i.e. devices without an entry at this device database shall be treated as untrusted.

Database is a descriptive term, which does not say how to implement it.

7.2.5.5.16.3

[GUIDELINE] If Bluetooth is supported, then the Bluetooth Device Database of an M-NCF should store the following information about an MHD device:

- Validity time of the entry from the time of its creation.

[Attributes]

S	A	n/a	n/a	M-NCF	n/a	VWW9M	

NOTE: The validity time specifies the duration for which temporary network access rights are granted.

7.2.5.5.16.4

[Guideline] If validity time is stored in the Bluetooth Device Database of a DLNA Device Class, then the validity time shall be specified in seconds.

[Attributes]

M	A	n/a	n/a	M-NCF	[80]	8OP4N	

7.2.5.5.17 NC M-NCF: Bluetooth Service Database Definition

7.2.5.5.17.1

[Guideline] If Bluetooth is supported, then an M-NCF shall maintain, or have full access to, a Bluetooth Service Database to store security-related information on services it supports.

[Attributes]

M	A	n/a	n/a	M-NCF	[80]	OQRJC	

7.2.5.5.17.2

[Guideline] If Bluetooth is supported, then the Bluetooth Service Database of an M-NCF shall, at a minimum, store the following information about a service they support:

- Authentication is required or not.
- Authorization is required or not.
- Encryption is required or not.

[Attributes]

M	A	n/a	n/a	M-NCF	[80] [81]	I7WWS	

7.2.5.5.17.3

[Guideline] If Bluetooth is supported, then an M-NCF shall create an entry in the Bluetooth Service Database for the PAN service.

[Attributes]

M	A	n/a	n/a	M-NCF	[81] [79]	QYXG5	

7.2.5.5.18 NC M-NCF: Bluetooth Establishing Network Access Rights

7.2.5.5.18.1

[Guideline] If Bluetooth is supported, then an M-NCF shall be able to initiate granting of network access rights to an MHD device by initiating a pairing procedure with the MHD device, as specified in 7.2.3.4.6.

[Attributes]

M	R	n/a	n/a	M-NCF	[82]	IVT6S	

NOTE: Network access rights are the pre-approval for future network connections from an MHD. Connection requests from an MHD with permanent network access rights are always granted. Connection requests from an MHD with temporary network access rights are granted if other additional conditions are satisfied.

Pairing can be initiated either by the MHD or the M-NCF.

7.2.5.5.18.2

[GUIDELINE] If Bluetooth is supported, then an M-NCF may initiate the pairing process to establish network access rights.

[ATTRIBUTES]

O	L	n/a	n/a	M-NCF	[82]	9YUGZ	

7.2.5.5.18.3

[GUIDELINE] If Bluetooth is supported, upon successful completion of the pairing procedure with an MHD, an M-NCF do one of following:

- Grant permanent network access rights to that MHD
- Grant temporary network access rights to that MHD
- Grant no access rights at all to that MHD

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	n/a	4BVTT	

7.2.5.5.18.4

[GUIDELINE] If Bluetooth is supported, in order for an M-NCF to grant permanent network access rights to an MHD, it shall

- Create an entry into the Bluetooth Device Database consisting of the BT device address, device name, and the associated link key
- Set the device as Trusted

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	[80]	47LYH	

NOTE: The choice is determined by implementation-specific mechanisms including, but not limited to, fixed rules and user interactions.

7.2.5.5.18.5

[GUIDELINE] If Bluetooth is supported, in order for an M-NCF to grant temporary network access rights to an MHD, it shall

- Create an entry into the Bluetooth device database consisting of the Bluetooth device address, device name, and the associated link key
- Set the device as Untrusted
- Set "Authorized for PAN" true
- Set the Validity time of the MHD device record, if the Validity time field is present

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	[80]	RX3WS	

NOTE: The choice of value for "Authorized for PAN service" is decided by implementation-specific mechanisms including, but not limited to, user interactions. When set to True, the connection request is granted after authentication. When set to False, the M-NCF can further initiate other mechanisms including, but not limited to, user interactions to grant or deny the connection request.

7.2.5.5.18.6

[Guideline] If Bluetooth is supported, in order for an M-NCF to grant neither permanent nor temporary network access rights to an MHD, it shall not create an entry into the Bluetooth device database for this MHD.

[Attributes]

M	A	n/a	n/a	M-NCF	[80]	X3WSS	

7.2.5.5.18.7

[Guideline] If Bluetooth is supported, an M-NCF should be able to grant one time network access to an MHD without granting permanent or temporary access rights to that MHD.

[Attributes]

S	A	n/a	n/a	M-NCF	n/a	7LYH2	

7.2.5.5.19 NC M-NCF: Bluetooth Managing Network Access Rights

7.2.5.5.19.1

[Guideline] If Bluetooth is supported, an M-NCF may convert temporary network access rights granted to an MHD to permanent network access rights by setting the "Trusted or Untrusted" field in the Bluetooth Device Database as "Trusted".

[Attributes]

O	A	n/a	n/a	M-NCF	[80]	BVTT4	

7.2.5.5.19.2

[Guideline] If Bluetooth is supported and if a validity time exists in the Bluetooth Device Database, an M-NCF may extend the duration of the validity time before the validity time expires.

[Attributes]

O	A	n/a	n/a	M-NCF	[80]	YUGZJ	

NOTE: This process can be triggered by implementation-specific mechanisms including, but not limited to, user requests.

7.2.5.5.20 NC M-NCF: Bluetooth Revoking Network Access Rights

7.2.5.5.20.1

[Guideline] If Bluetooth is supported, an M-NCF shall be able to revoke permanent or temporary network access rights of an MHD in the Bluetooth Device Database at any time.

[Attributes]

M	A	n/a	n/a	M-NCF	[79]	VT6S9	

NOTE: This process can be triggered by implementation-specific mechanisms including, but not limited to, user requests.

7.2.5.5.20.2

[Guideline] If Bluetooth is supported and if a validity time exists in the Bluetooth Device Database, an M-NCF should revoke temporary network access rights from an MHD when the validity time of that MHD expires.

[ATTRIBUTES]

S	A	n/a	n/a	M-NCF	n/a	YXG55	

NOTE: This process can be triggered by implementation-specific mechanisms including, but not limited to, user requests.

7.2.5.5.20.3

[GUIDELINE] If Bluetooth is supported, in order for an M-NCF to revoke network access rights from an MHD, it shall do so by deleting the entry of the MHD from the Bluetooth Device Database, and terminating the MHD connection, if the connection is present.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	[80]	8IWAN	

7.2.5.5.21 NC M-NCF: Bluetooth Access Control

7.2.5.5.21.1

[GUIDELINE] If Bluetooth is supported, upon receiving a connection request from an MHD, an M-NCF shall accept or deny the connection request according to the following guidelines 7.2.5.5.21.2 to 7.2.5.5.21.5.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	n/a	7WWS5	

7.2.5.5.21.2

[GUIDELINE] If the PAN service record in the Bluetooth Service Database has the value of "Authorization required" set to False, and "Authentication required" is set to False, an M-NCF shall accept the connection request from the MHD.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	[80] [81]	QRJCP	

7.2.5.5.21.3

[GUIDELINE] If the PAN service record in the Bluetooth service database has the value of "Authorization required" set to False, and "Authentication required" set to True, the connection request shall be accepted by an M-NCF after authentication is completed successfully as specified in 7.2.3.4.6.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	[80] [81]	OP4NB	

7.2.5.5.21.4

[GUIDELINE] If the PAN service record in the Bluetooth service database has the value of "Authorization required" set to True, then the connection request from MHD shall be accepted by the DLNA Device Class in the following cases:

- If there is a record of the MHD in the Bluetooth device database and the device is marked as Trusted, the connection request shall be accepted by the device after authentication is completed successfully as specified in 7.2.3.4.6.
- If there is a record of the MHD in the Bluetooth device database, the device is marked as Untrusted, and has the value of "Authorized for PAN service" set to True, the connection

request shall be accepted by the device after authentication is completed successfully as specified in 7.2.3.4.6.

[Attributes]

M	A	n/a	n/a	M-NCF	[80] [81]	WW9MI	

7.2.5.5.21.5

[Guideline] If the PAN service record in the Bluetooth service database has the value of "Authorization required" set to True, and the Bluetooth Device Database does not have a record of the MHD requesting the connection, then an M-NCF may use other implementation-specific mechanisms to determine whether to accept or deny the connection request after authentication is completed as specified in 7.2.3.4.6.

[Attributes]

O	A	n/a	n/a	M-NCF	[80] [81]	V2GQ8	

7.2.5.5.22 NC M-NCF Link Key Usage and Management

7.2.5.5.22.1

[Guideline] If Bluetooth is supported and there is a record of the MHD in a DLNA Device Class's Bluetooth Device Database, then the MHD Device and the DLNA Device Class shall use the link key stored in the database for authentication and deriving the encryption key for this session with the MHD.

[Attributes]

M	A	n/a	n/a	M-NCF	[80] [81]	2GQ88	

NOTE: Implementation-specific mechanisms include, but are not limited to, fixed rules, user interaction or other trusted external entities. A trusted external entity includes a user with a valid PIN or other implementation specific applications, e.g. through network management applications.

7.2.5.5.22.2

[Guideline] If Bluetooth is supported and there is no record of the MHD in a DLNA Device Class's Bluetooth Device Database, then the device may create a temporary link key and distribute it to the MHD. This temporary link key is used for authentication and deriving the encryption key.

[Attributes]

O	A	n/a	n/a	M-NCF	[80] [81]	W9MI3	

7.2.5.5.23 NC M-NCF: BT PIN Type

[Guideline] If Bluetooth is supported, then an M-NCF should implement support for variable PIN codes.

[Attributes]

S	L	n/a	n/a	M-NCF	[82]	P4NBO	

NOTE: M-NCF can support Fixed BT PIN type, but it is encouraged to support variable PIN to enhance security.

7.2.5.5.24 NC M-NCF: Bluetooth PS support

7.2.5.5.24.1

[Guideline] If Bluetooth is supported, then an M-NCF shall support the 'Sniff' Bluetooth power savings mode.

[ATTRIBUTES]

M	R	n/a	n/a	M-NCF	[82]	RJCPV	

7.2.5.5.24.2

[GUIDELINE] If Bluetooth is supported, then an M-NCF shall accept at any time all MHD's requests related to setting the Bluetooth link between them in 'Sniff' mode.

[ATTRIBUTES]

M	L	n/a	n/a	M-NCF	[82]	WWS55	

7.2.5.5.25 NC M-NCF: NC-PS Mode Transitions with Bluetooth

7.2.5.5.25.1

[GUIDELINE] If Bluetooth is supported, then an M-NCF shall decide that the transition from the 'Active' to 'Standby' NC-PS mode is completed when the Bluetooth process to put the link from 'Active' to 'Sniff' Bluetooth mode is completed successfully.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	[82]	XG55W	

NOTE: While the NC-PS mode is 'Standby', the MHD can request at any time to change the parameters of the 'Sniff' Bluetooth power saving mode. This does not trigger a transition out of the 'Standby' NC-PS mode.

The moment the M-NCF decides that the transition from 'Active' to 'Standby' NC-PS mode is completed, it needs to start any traffic reduction operations as described in 7.2.5.5.9.

7.2.5.5.25.2

[GUIDELINE] If Bluetooth is supported, then an M-NCF shall decide that the transition from the 'Standby' to 'Active' NC-PS mode is completed when the Bluetooth process to put the link from 'Sniff' to 'Active' mode is completed successfully.

[ATTRIBUTES]

M	A	n/a	n/a	M-NCF	[82]	T6S95	

NOTE: The moment the M-NCF decides that the transition from 'Standby' to 'Active' NC-PS mode is completed, it can cease any traffic reduction operations as described in 7.2.5.5.9.

7.2.5.5.26 NC M-NCF: BT-Ethernet DLNAQOS Access Category Mapping

7.2.5.5.26.1

[GUIDELINE] If DLNAQOS is supported and an M-NCF bridges Bluetooth in the mobile connectivity domain and Ethernet in the home connectivity domain, then the device should maintain unmodified the IEEE 802.1Q headers and DSCP tags, when traffic is received from one connectivity domain and transmitted to the other.

[ATTRIBUTES]

S	A	n/a	n/a	M-NCF	[81]	UGZJX	

7.2.5.5.27 NC M-NCF: BT-802.11 DLNAQOS Access Category Mapping

7.2.5.5.27.1

[GENERAL] The BNEP protocol is designed to encapsulate Ethernet packets over BT. Therefore, the BT priority taggings are the same as in Ethernet.

7.2.5.5.27.2

[GUIDELINE] If DLNAQOS is supported and an M-NCF bridges Bluetooth in the mobile connectivity domain and 802.11 in the home connectivity domain, then packets received on the 802.11 interface and transmitted on the Bluetooth interface should include the IEEE 802.1D user priority value in IEEE 802.1Q header and the DSCP tag corresponding to the WMM Access Category of the received 802.11 packets in accordance with:

Table 12 — BT-802.11 DLNAQOS Access Category Mapping

WMM Access Category	IEEE 802.1D priority		DSCP
AC_BK	1	BK	0x08
AC_BE	0	BE	0x00
AC_VI	5	VI	0x28
AC_VO	7	NC	0x38

7.2.5.5.27.3

[GUIDELINE] If DLNAQOS is supported and an M-NCF bridges Bluetooth in the mobile connectivity domain and 802.11 in the home connectivity domain, then packets received on the Bluetooth interface and transmitted on the 802.11 interface should include the WMM Access Category corresponding to the IEEE 802.1D user priority value in IEEE 802.1Q header tag of the received 802.3 packets in accordance with

Table 13 — IEEE 802.1D User Priority Values

IEEE 802.1D priority		DSCP	WMM Access Category
1	BK	0x08	AC_BK
2	-	0x10	
0	BE	0x00	AC_BE
3	EE	0x18	
4	CL	0x20	AC_VI
5	VI	0x28	
6	VO	0x30	AC_VO
7	NC	0x38	

[ATTRIBUTES]

S	A	n/a	n/a	M-NCF	[81] [1] [2] [85] [32]	VTT4P

7.2.5.5.27.4

[GUIDELINE] If DLNAQOS is supported and an M-NCF bridges Bluetooth in the mobile connectivity domain and 802.11 in the home connectivity domain, if a packet received on the Bluetooth interface does not contain an 802.1Q tag, the M-NCF should look at the DSCP tag and map that to a WMM Access Category in accordance with the table in 7.2.5.5.27.3 and preserve the DSCP tag across Bluetooth and 802.11 segments.

[ATTRIBUTES]

S	A	n/a	n/a	M-NCF	[81] [1] [85] [32]	LYH2P

7.2.5.5.27.5

[GUIDELINE] If DLNAQOS is supported and an M-NCF bridges Bluetooth in the mobile connectivity domain and 802.11 in the home connectivity domain, if a packet received on the Bluetooth interface contains neither an 802.1Q tag or a DSCP tag, the packet should be passed through to the 802.11 interface without the addition of any priority tagging.

[ATTRIBUTES]

S	A	n/a	n/a	M-NCF	[81] [1] [85] [32]	3WSSV

7.2.5.5.27.6

[GUIDELINE] If DLNAQOS is supported and an M-NCF bridges Bluetooth in the mobile connectivity domain and 802.11 in the home connectivity domain, and a packet received on the 802.11 interface is not tagged with a WMM Access Category, then the packet should be passed through to the Bluetooth interface without the addition of any priority tagging.

[ATTRIBUTES]

S	A	n/a	n/a	M-NCF	[81] [1] [85] [32]	WSSV9

7.3 Device Discovery and Control

7.3.1 General

This subclause of the DLNA Home Networked Device Interoperability Guidelines covers the guidelines for implementing device discovery and control using the UPnP device architecture. These guidelines balance the needs for both devices and control points, and specify rules for a variety of protocol areas, such as SSDP, GENA events, SOAP actions, and HTTP transports for *UPnP communications*. It should be noted that HTTP guidelines in this subclause apply only to UPnP-related transactions and not to content transfer transactions.

In this subclause, the following terms are used.

- UPnP endpoints: Refers to both UPnP devices and UPnP control points.
- HTTP clients: Refers to the HTTP clients used for *UPnP communications*. HTTP client guidelines in this subclause do not apply to HTTP transport for content transfers or playback.
- HTTP servers: Refers to the HTTP servers used for *UPnP communications*. HTTP server guidelines in this subclause do not apply to HTTP transport for content transfers or playback.

The general rules for handling XML documents and fragments are specified in 6.7.

7.3.2 Device Discovery and Control Guidelines

7.3.2.1 DDC UPnP Device Architecture

7.3.2.1.1

[GUIDELINE] DLNA Device Classes and Device Capabilities shall fully support the applicable mandatory portions of the UPnP Device Architecture v1.0 (UPnP DA) for discovery, description, control, eventing, and presentation.

[ATTRIBUTES]

M	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[47]	YH2P3	A

NOTE: DLNA specifies UPnP Device Architecture 1.0 (UPnP DA) as the basic protocol framework for Device Classes.

7.3.2.1.2

[GUIDELINE] A UPnP control point designed for a version of a UPnP Device Architecture, shall also be able to interoperate with later versions of the UPnP Device Architecture that have the same major version.

"Interoperate" means that a control point that has certain capabilities for older devices can at least provide the same capabilities for newer devices. For example, a control point that can discover an older UPnP device, parse its device and service description files, and invoke its UPnP actions shall be able to do those same things with a UPnP device with a newer minor revision of the UPnP Device Architecture.

[ATTRIBUTES]

M	C	DMP DMC M-DMP +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMC M-DMU, M-DMD	MIU	[47]	TT4PY	A

NOTE: Clause 1 of the UPnP Device Architecture 1.0 indicates that advances in minor version of the UPnP Device Architecture are a superset of earlier (minor) versions with the same major version. This means that future UPnP devices with a newer minor revision will implement all of the behavior required of the previous device architectures.

Although not explicitly stated by the UPnP Device Architecture, the intent of such backwards compatibility rules is to enable forward compatibility of control points with newer, minor revisions of the UPnP device architecture. Guidelines 7.3.2.1.2 and 7.3.2.1.3 formally require forward compatibility of control points, which is necessary for future interoperability.

Note that a version of the UPnP Device Architecture appears in the <specVersion> element of the device and service descriptions and the SERVER header in SSDP, SOAP, and GENA messages.

One way to implement 7.3.2.1.2 is for the UPnP control point to treat the minor version a UPnP device architecture as 0 (i.e. ignore the minor version).

7.3.2.1.3

[GUIDELINE] A UPnP control point designed for a version of a UPnP device type or service type shall be able to interoperate with later versions of the same device type or service type.

"Interoperate" means that a control point that has certain capabilities for an older device can at least provide the same capabilities for a newer device of the same type and services. For example, a control point that can discover an older UPnP AV MediaServer and invoke its CDS:Browse action shall be able to discover and invoke CDS:Browse on a newer UPnP MediaServer. The newer UPnP device can be newer because the device type and/or one of its associated UPnP services are of a newer version.

[ATTRIBUTES]

M	C	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	GZJXU	A

NOTE: 2.1 of the UPnP Device Architecture 1.0 states that standardized device types and service types are required to be a superset of all previous versions of the same device/service type. This means that future UPnP device and service types will require all of the behavior defined for previous versions.

Note that a device version appears as part of the value in a <deviceType> element of a device description file. Similarly, the service version appears as part of the value in a <serviceType> element of a service description file. Version numbers also appear in NT, ST, and USN headers of SSDP messages. Furthermore, a service version appears in the xmlns namespace attributes of SOAP messages.

Older control points that interact with newer UPnP device/service types are not expected to interoperate with conventions established for the newer device or service type, but they will still use parts of the UPnP device/service that are compatible with the older conventions.

7.3.2.1.4

[GUIDELINE] If a SOAP action was defined in the specification of a previous service version, a UPnP control point may specify the xmlns namespace attribute for the service type and the SOAPACTION header in the SOAP request with the earlier service version.

[ATTRIBUTES]

O	C	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	6S95W	A

NOTE: In other words, if a SOAP action was defined in the specification of a previous service version (e.g. version 1), then a UPnP control point can invoke the SOAP action with the earlier service version regardless of the service version described in the device description.

7.3.2.2 DDC UPnP Auto IP Support

7.3.2.2.1

[GUIDELINE] UPnP devices and control points shall implement the Auto-IP behavior defined in [47] even if they implement a DHCP server.

[ATTRIBUTES]

M	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[47]	G55WU	A

NOTE: DLNA Device Classes that do not properly support DHCP and AutoIP as required by the UPnP DA can cause IP addressing problems for other UPnP entities.

7.3.2.2.2

[GUIDELINE] Whenever a UPnP device switches to a new IP address (whether assigned through Auto-IP or DHCP), the device should send an ssdp:byebye message for (and on) the old IP address.

For (and on) the old IP address means that the IP address indicated in the (UDP header of the) ssdp:byebye matches the old IP address of the UPnP device.

[ATTRIBUTES]

S	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	WANFQ	

NOTE: This allows control points that discovered the UPnP device on the old IP address to know that the UPnP device is no longer available at the old address. However, this behavior is not always possible for implementations built on some platforms.

7.3.2.2.3

[GUIDELINE] If UPnP devices and control points use a self-assigned IP address, then they shall implement duplicate address detection before assigning the address.

[ATTRIBUTES]

M	R	HND +DN+ +PU+ +UP+ +PR1+ +PR2+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[47]	WS55I	A

NOTE: This guideline repeats a UPnP DA requirement that prevents the assignment of conflicting IP addresses (see subclauses 0.2 and 0.3 of [47]).

7.3.2.2.4

[GUIDELINE] If a DLNA device class implements a DHCP server, it shall provide a mechanism to disable and enable the DHCP server.

[ATTRIBUTES]

M	A	HND	MHD	MIU	[47]	JCPVD	

NOTE: This guideline clarifies the condition for DHCP server support in a DLNA device. The user needs to be able to disable the DHCP server function to avoid the presence of multiple DHCP servers providing different network configurations on the same home network.

7.3.2.3 DDC UPnP SSDP Default Port

7.3.2.3.1

[GUIDELINE] UPnP devices shall receive and process M-SEARCH messages on port 1900.

[ATTRIBUTES]

M	C	DMS DMR DMPr	M-DMS	MIU	[47]	4NBO8	C

NOTE: This requirement ensures that devices always listen on port 1900. Devices respond to M-SEARCH messages according to [47].

7.3.2.3.2

[GUIDELINE] UPnP control points should receive and process NOTIFY messages on port 1900.

[ATTRIBUTES]

S	C	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMD M-DMU	MIU	[47]	DEPDO	N

NOTE: This guideline encourages UPnP control points to listen on port 1900 and use the information from NOTIFY messages (ssdp:alive and ssdp:byebye). However, some UPnP control points rely on M-SEARCH messages instead of NOTIFY messages to keep track of UPnP devices. For example, if UPnP control points connect sporadically to the network to perform media-related tasks.

7.3.2.3.3

[GUIDELINE] UPnP devices shall always explicitly specify port 1900 in every HOST header tag for every SSDP message.

[ATTRIBUTES]

M	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	9MI39	

7.3.2.3.4

[GUIDELINE] UPnP control points receiving an SSDP message without the port number in the HOST header tag, shall infer the port number is 1900.

[ATTRIBUTES]

M	R	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	GQ888	A

7.3.2.3.5

[GUIDELINE] UPnP control points shall send M-SEARCH messages using a source port greater than 1024 and not 1900.

[ATTRIBUTES]

M	L	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	Q888R	A

NOTE: These guidelines are based on a Microsoft technical advisory regarding security concerns for UPnP.

7.3.2.3.6

[GUIDELINE] UPnP devices may ignore M-SEARCH messages if the originating source port is less than or equal to 1024 or equal to 1900.

[ATTRIBUTES]

O	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	MI39Y	

7.3.2.4 DDC UPnP Discovery Robustness

7.3.2.4.1

[GUIDELINE] UPnP endpoints (devices and control points) should wait a random amount of time, between 0 and 100 milliseconds after acquiring a new IP address, before sending advertisements or initiating searches on a new IP interface.

[ATTRIBUTES]

S	L	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[47]	NBO8C	A

NOTE: This suggestion avoids SSDP discovery flooding on home networks that contain a large number of UPnP endpoints.

7.3.2.4.2

[GUIDELINE] UPnP network devices shall not send more than 10 ssdp:alive messages on a single network interface in any given 200 ms period.

[ATTRIBUTES]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	CPVDX	

NOTE: This guideline prevents lost packets caused by buffer overflow of Ethernet drivers by UPnP devices with many services or embedded devices in the device hierarchy.

7.3.2.4.3

[GUIDELINE] UPnP devices shall send each *advertisement set* more than once on a single network interface (It is recommended that UPnP devices send a total of 2 or 3 *advertisement sets*).

An *advertisement set* refers to the set of 3+2d+k ssdp:alive messages that UPnP device sends as part of its periodic advertisements.

The repeated *advertisement set*s are referred to as *duplicate sets*.

The transmission windows for *advertisement set*s and *duplicate sets* cannot overlap in time.

[ATTRIBUTES]

M	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	S55IQ

NOTE: This guideline clarifies how a UPnP device needs to retransmit its advertisements. However, implementers are reminded that advertising too frequently runs the risk of flooding the SSDP channel.

7.3.2.4.4

[GUIDELINE] A UPnP device that uses the same UDN on multiple network interfaces, shall send each individual ssdp:alive message (from an *advertisement set*) on all interfaces within a 10 second transmission window.

Time intervals between individual ssdp:alive messages on a single interface are not restricted by this requirement.

[ATTRIBUTES]

M	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	ANFQT

NOTE: Control points need a way to determine the most reliable network route to the UPnP device. This guideline ensures that control points will receive an individual ssdp:alive message on all network interfaces within a 10 second transmission window.

7.3.2.4.5

[GUIDELINE] The interval of sending these *advertisement groups* on a single network interface shall be less than ½ the CACHE-CONTROL value.

The first *advertisement set* and the *duplicate sets* (transmitted on a single network interface) make up an advertisement group.

[ATTRIBUTES]

M	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	55WUY

NOTE: For consistency and interoperability, devices need to advertise more often than their notification cycle. However, implementers are reminded that advertising too frequently runs the risk of flooding the SSDP channel.

7.3.2.4.6

[GUIDELINE] The CACHE-CONTROL value should be at least 1800, as recommended in the UPnP device architecture.

[ATTRIBUTES]

S	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	S95WV

NOTE: Most devices that remain on the network for long periods have CACHE-CONTROL value of 1800. However, some devices (mobile, wireless, etc.) might want a smaller CACHE-CONTROL value.

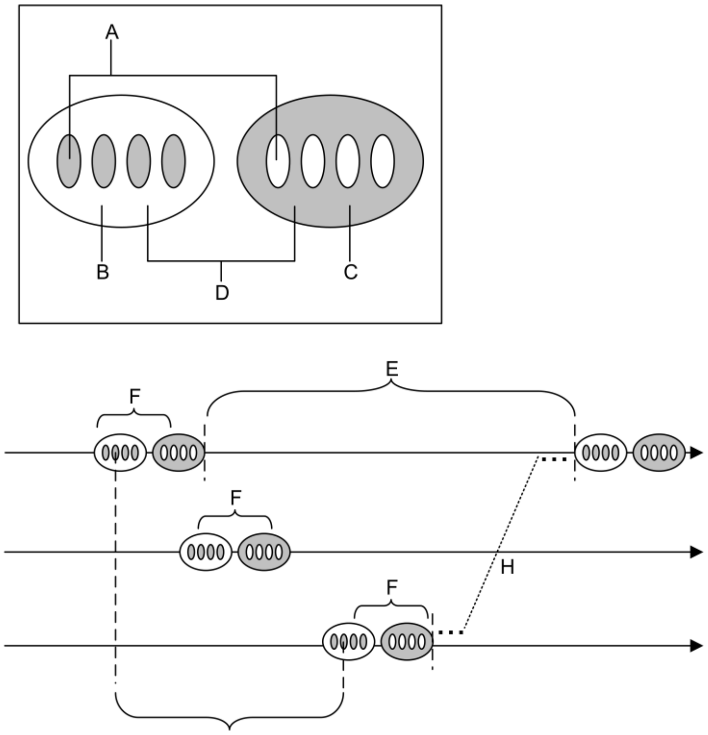


Figure 27 — UPnP Discovery Robustness

(a).One or more ssdp:alive messages, within *advertisement sets* and *duplicate sets*.

(b).Advertisement set of 3+2d+k ssdp:alive messages.

(c). Duplicate set of 3+2d+k ssdp:alive messages. (see 7.3.2.4.3)

(d).Combined *advertisement set* and *duplicate sets* make an advertisement group.(see 7.3.2.4.5)

(e).Delay between *advertisement groups* on same network is less than ½ of CACHE-CONTROL value. (see 7.3.2.4.5)

(f). Any arbitrary window of 200 ms have 10 or fewer ssdp:alive messages. (see 7.3.2.4.2) An entire *advertisement set* need not fit inside the 200 ms window.

(g).An individual ssdp:alive message shall have all corresponding ssdp:alive sent within a 10 second transmission window. (see 7.3.2.4.4)

(h).This delay is not drawn to scale.

7.3.2.4.7

[GUIDELINE] Due to the unreliable nature of UDP, control points should send each M-SEARCH message more than once, not to exceed 10 M-SEARCH requests in a 200 ms period. An

M-SEARCH message and its repeated duplicates should all be sent within a 10 second period.

[ATTRIBUTES]

S	R	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	ZJXUT	A

NOTE: Wireless access points do not retry multicast traffic and can cause UPnP discovery problems. This recommendation repeats advice from the UPnP DA.

The 10 M-SEARCH messages per 200 ms period is consistent with maximum saturation limit of 10 ssdp:alive messages per 200 ms period for UPnP devices (7.3.2.4.2). Likewise, the sending all of the M-SEARCH messages in a window of 10 seconds is consistent with the requirement where all duplicates of an individual ssdp:alive message are sent within 10 seconds (7.3.2.4.4).

7.3.2.4.8

[GUIDELINE] The control point should wait at least the amount of time specified in the MX header for responses to arrive from devices. The time waited for responses should be extended by additional time (a second or two) to allow for network propagation and processing delays.

[ATTRIBUTES]

S	R	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	T4PYW	A

7.3.2.4.9

[GUIDELINE] Upon startup, UPnP devices should broadcast an ssdp:byebye before sending the initial ssdp:alive onto the local network.

[ATTRIBUTES]

S	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	H2P3Y	

NOTE: The UPnP device architecture specification does not account for devices that reset without sending an ssdp:byebye. If devices do not send an ssdp:byebye when returning to the network after such an event, control points cannot tell if the received announcement is for a new device instance, or is merely a periodic announcement for the same device instance.

Sending an ssdp:byebye as part of the normal start up process for a UPnP device ensures that UPnP control points with information about the previous device instance will safely discard state information about the previous device instance before communicating with the new device instance.

7.3.2.4.10

[GUIDELINE] UPnP control points after acquireing a new IP address shall initiate searches (e.g. M-SEARCH) on the new IP interface.

[ATTRIBUTES]

M	A	DMP DMC +UP+ +PR1+ +PR2+ +DN+ +PU+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	SSV9L	A

NOTE: This guideline improves the user experience by allowing a UPnP control point (e.g. DMP) to discover UPnP devices (e.g. DMSs) more quickly after a UPnP control point and devices transition from a DHCP to an Auto IP address or vise versa. Otherwise a UPnP control point might have to wait until the next set of advertisements (i.e. NOTIFY packets) from a UPnP device to discover it, which could be a few minutes depending upon the CACHE-CONTROL max-age value. The timing of issuing M-SEARCH is as specified in guideline 7.3.2.4.1 as guidance.

7.3.2.5 DDC UPnP HTTP Support and General Rules

7.3.2.5.1

[GUIDELINE] UPnP endpoints (devices and control points) shall support at least HTTP/1.0 ([21]) for performing *UPnP communications*, excluding SSDP communications.

For SSDP communications, UPnP endpoints shall use the HTTP/1.1 message format defined in [47].

[ATTRIBUTES]

M	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[21] [47]	SV9LG	A

NOTE: SSDP messages are based on the HTTP/1.1 message format with method and header extensions.

7.3.2.5.2

[GUIDELINE] UPnP devices shall support HTTP/1.1.

[ATTRIBUTES]

M	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[33] [47]	2P3YZ	

NOTE: Although HTTP/1.0 is the baseline for *UPnP communications*, HTTP/1.1 is encouraged.

7.3.2.5.3

[GUIDELINE] HTTP servers of UPnP control points shall support HTTP/1.1.

[ATTRIBUTES]

M	R	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[33] [47]	4PYWQ	A

7.3.2.5.4

[GUIDELINE] HTTP clients of UPnP control points should use and support HTTP/1.1.

[ATTRIBUTES]

S	C	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[33] [47]	JXUT6	A

7.3.2.5.5

[GUIDELINE] The message format of HTTP responses (sent by HTTP servers of both devices and control points) shall be compliant with the version number specified by the request.

[ATTRIBUTES]

M	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[23]	95WVR	A

NOTE: The clarifying IETF specification ([23]) states that HTTP/1.1 servers should return HTTP/1.1 even if the HTTP server receives a request marked with HTTP/1.0. The robustness rules, specified by the HTTP specification, enables clients and servers that employ different HTTP version numbers to coexist properly.

7.3.2.5.6

[GUIDELINE] HTTP/1.1 servers of UPnP endpoints (devices and control points) should return HTTP version 1.1 in the response header, regardless of the version specified in the HTTP client's request.

[ATTRIBUTES]

S	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[23]	5WUY7	A

7.3.2.5.7

[GUIDELINE] HTTP servers of UPnP endpoints (devices and control points) shall not report a higher version of HTTP than is actually supported by the implementation.

[ATTRIBUTES]

M	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	NFQTX	A

7.3.2.5.8

[GUIDELINE] The HTTP servers and clients of UPnP endpoints (devices and control points) shall be able to properly parse all HTTP headers provided to them. In particular, they shall support HTTP header tags in any order and accept the tag name in a case insensitive manner and associated data in a case sensitive manner. If a header tag is not recognized by a UPnP endpoint, it shall ignore the header and continue parsing the packet.

This guideline applies to all HTTP headers, regardless of whether the DLNA guidelines define a BNF syntax for the HTTP header value. In other words, all endpoints shall implement a "parse and interpret" or "parse and ignore" when parsing an HTTP header field.

[ATTRIBUTES]

M	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[21] [33]	55IQI	A

NOTE: This guideline specifies a minimal robustness level for parsing HTTP headers. The HTTP headers include both HTTP headers defined in [33] and other headers, such as DLNA defined and vendor defined headers, used for DLNA interoperability.

7.3.2.5.9

[GUIDELINE] The HTTP servers and clients of UPnP endpoints (devices and control points) shall include the Content-Type header tag in every UPnP-related TCP-based HTTP transaction (SOAP, GENA, and device/service description) that contains an XML body. This content type shall always be marked as the following:

- text/xml; charset="utf-8"

Note that charset parameter value is case insensitive and double quotations may be omitted.

Furthermore, the XML shall be encoded in UTF-8.

UPnP endpoints (devices and control points) that receive a content type of text/xml shall infer UTF-8 character set encoding.

[ATTRIBUTES]

M	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[21] [33]	PVDXI	A

NOTE: Restricting *UPnP communications* to UTF-8 simplifies implementations and makes it so that devices need not implement a separate parsing engine for every local region.

7.3.2.5.10

[GUIDELINE] If the DLNA guidelines define a BNF syntax for an HTTP header, then the HTTP servers and clients of UPnP endpoints shall not include white spaces in the header-value of HTTP headers unless SP and LWS are explicitly specified in the syntax (BNF) definitions.

If the DLNA guidelines do not define a BNF syntax for an HTTP header, then the header shall conform to the message-header syntax in subclause 4.2 of [33], regardless of whether the HTTP header is defined in [33] or if the HTTP header is vendor-defined. Note that the syntax for field-value permits LWS to separate tokens and other data in the field-value.

Implied LWS between the HTTP header-name and the HTTP header-value are permitted as specified in [33], regardless of whether the DLNA guidelines specify a BNF syntax for the HTTP header.

[ATTRIBUTES]

M	A	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	BO8CA	A

NOTE: Conformance to subclause 6.4, the restriction on the use of SP and LWS characters is applied in UPnP HTTP communications. Also note that white spaces between header-name and header-value are still acceptable.

The header-name and header-value are defined by the field-name and field-value tokens of the message-header syntax in subclause 4.2 of [33].

7.3.2.6 DDC UPnP HTTP/1.0 Rules

7.3.2.6.1

[GUIDELINE] For all HTTP/1.0 transactions, the HTTP server shall close the TCP connection after sending the complete HTTP response. This guideline covers both kinds of HTTP/1.0 transactions:

- HTTP/1.1 server responds to an HTTP/1.0 request, and
- HTTP/1.0 server responds to an HTTP/1.1 request.

[ATTRIBUTES]

M	C	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[21] [33]	I39YP	A

NOTE: The use of the Content-Length field greatly reduces the parsing complexity of HTTP message bodies on a UPnP control point.

When an HTTP server responds to an HTTP/1.0 request without closing the socket, the Content-Length field is the only method that client can use to determine that the entire response was received.

7.3.2.6.2

[GUIDELINE] If a UPnP device's HTTP server responds to a SOAP request as part of an HTTP/1.0 transaction, then the UPnP device shall close the TCP connection after the response has been sent.

This guideline covers both kinds of HTTP/1.0 transactions:

- HTTP/1.1 server responds to an HTTP/1.0 request, and
- HTTP/1.0 server responds to an HTTP/1.1 request.

Also note that in both of the above cases, the UPnP device has the HTTP server and the UPnP control point issues the HTTP request.

[ATTRIBUTES]

M	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[21] [33]	39YPQ	

NOTE: This is the proper behavior for a UPnP device, as it follows standard HTTP rules.

7.3.2.6.3

[GUIDELINE] If a UPnP control point's HTTP server responds to a GENA event as part of an HTTP/1.0 transaction, then the control point shall close the TCP connection after the response has been sent.

This guideline covers both kinds of HTTP/1.0 transactions:

- HTTP/1.1 server responds to an HTTP/1.0 request, and
- HTTP/1.0 server responds to an HTTP/1.1 request.

Also note that in both of the above cases, the UPnP control point has HTTP server and the UPnP devices issues the HTTP request.

[ATTRIBUTES]

M	R	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[21] [33]	88RDT	A

NOTE: This is the proper behavior for a UPnP control point.

7.3.2.7 DDC UPnP HTTP/1.1 Transaction Rules

7.3.2.7.1

[GUIDELINE] A UPnP device's HTTP server shall close the TCP connection after responding to a SOAP request with the CONNECTION: CLOSE token.

[ATTRIBUTES]

M	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[33]	O8CA7	

7.3.2.7.2

[GUIDELINE] A UPnP control point's HTTP server shall close the TCP connection after responding to an event that was sent with the CONNECTION: CLOSE token.

[ATTRIBUTES]

M	R	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[33]	VDXID	A

7.3.2.7.3

[GUIDELINE] HTTP clients of UPnP endpoints (devices and control points) shall not report support for HTTP/1.1 unless they also support *Chunked Transfer Coding* and correctly parse a 100 (Continue Response), as required by the HTTP/1.1 specification.

[ATTRIBUTES]

M	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	5IQI5	A

NOTE: Only HTTP clients that support *Chunked Transfer Coding* and 100 (Continue Response) messages can initiate HTTP/1.1 transactions.

7.3.2.7.4

[GUIDELINE] The HTTP servers of UPnP endpoints (devices and control points) shall use the Content-Length HTTP header tag at all times, unless the connection will be closed after the response is sent or *Chunked Transfer Coding* is used.

[ATTRIBUTES]

M	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	FQTXQ	A

NOTE: When an HTTP/1.1 server sends a response back to the client without closing the socket afterwards, the client will not know when the entire response was received, unless the response was encoded with *Chunked Transfer Coding*, without interpreting the Content-Length header.

7.3.2.7.5

[GUIDELINE] The HTTP clients of UPnP endpoints (devices and control points) may issue HTTP/1.1 requests encoded with *Chunked Transfer Coding*.

[ATTRIBUTES]

O	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	WUY7P	A

NOTE: These guidelines repeat the HTTP specification by noting the permitted use of *Chunked Transfer Coding* for HTTP/1.1 requests. HTTP clients of UPnP devices can use *Chunked Transfer Coding* for delivery of UPnP GENA events. HTTP clients of UPnP control points can use *Chunked Transfer Coding* for delivery of UPnP SOAP actions. As such, HTTP/1.1 servers of UPnP endpoints are required to support HTTP/1.1 requests encoded with *Chunked Transfer Coding*.

7.3.2.7.6

[GUIDELINE] The HTTP servers of UPnP endpoints (devices and control points) shall accept, decode, and respond to HTTP/1.1 requests encoded with *Chunked Transfer Coding*.

[ATTRIBUTES]

M	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	5WVR8	A

7.3.2.8 DDC UPnP HTTP Persistent Connections

[GUIDELINE] This guideline no longer applies.

[ATTRIBUTES]

		n/a	n/a	n/a	n/a	XUT6R	D

7.3.2.8.1

[GUIDELINE] The HTTP clients and servers of UPnP endpoints (devices and control points) should support persistent HTTP/1.1 connections and pipelining.

[ATTRIBUTES]

S	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	PYWQP	A

NOTE: Persistent HTTP connections allow devices and control points to use fewer resources when communicating. Pipelining adds the ability for control points to queue requests onto an existing session.

7.3.2.8.2

[GUIDELINE] The HTTP clients of UPnP endpoints should use persistent HTTP/1.1 connections.

[ATTRIBUTES]

S	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	P3YZ8	A

NOTE: The default behavior for HTTP/1.1 is a persistent connection. Persistent connections result in no accumulation of TCP TIME-WAIT because the originator of the connection closes the socket.

7.3.2.8.3

[GUIDELINE] The HTTP clients of UPnP endpoints shall fall back to non-pipelining if the connection is closed after the first request and a second (or more) request from the same network entity is pending.

[ATTRIBUTES]

M	L	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	V9LGQ	A

NOTE: This guideline ensures consistent and correct behavior between mixes of UPnP endpoints that might or might not support HTTP pipelining.

7.3.2.8.4

[GUIDELINE] The HTTP servers of UPnP endpoints (devices and control points) that do not support persistent connections shall answer the first HTTP request from the requesting UPnP control point and close the TCP connection to correctly ignore other requests.

[ATTRIBUTES]

M	C	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	9LGQX	A

7.3.2.8.5

[GUIDELINE] The HTTP clients of UPnP endpoints that send multiple requests in a single HTTP session shall be ready to open new HTTP sessions if the device does not respond to all requests on the initial HTTP session.

[ATTRIBUTES]

M	C	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	3YZ8Z	A

7.3.2.8.6

[GUIDELINE] The HTTP clients of UPnP endpoints shall close a persistent connection (HTTP/1.1) within 60 seconds of inactivity (i.e., no traffic and no pending requests).

This guideline applies to both UPnP devices and control points. Context of this guideline is specific to UPnP-related communications, excluding SSDP communications. This guideline does not apply to the transport layer communications for media content transfers.

[ATTRIBUTES]

M	L	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	YWQPE	A

NOTE: This prevents control points and devices from holding network sockets for an unnecessarily long period.

7.3.2.9 DDC UPnP Device Responsiveness

7.3.2.9.1

[GENERAL] UPnP Device Architecture specification requires UPnP devices to complete the SOAP response in 30 seconds. However, this can be difficult to guarantee at the implementation layer for all types of UPnP actions. These guidelines attempt to strike a balance between ideal goals and practical implementation needs for both devices and control points.

That being stated, the original inspiration for these guidelines is that some UPnP AV MediaServer devices cannot guarantee that a response will complete within 30 seconds for a variety of reasons. Network bandwidth, query complexity, and hardware performance can vary. This being the case, such devices shall still begin their response within 30 seconds.

Also note that a UPnP AV MediaServer can reduce a long transmission time for a SOAP response (for a CDS:Browse or CDS:Search action) by reducing the number of returned items in the result. See guideline 7.4.1.4.10 for more information.

7.3.2.9.2

[GUIDELINE] UPnP devices shall begin the transmission of a SOAP response within 27 seconds of receiving a complete SOAP request.

[ATTRIBUTES]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	UT6RC	

7.3.2.9.3

[GUIDELINE] UPnP devices should begin the transmission of SOAP responses as soon as possible.

[ATTRIBUTES]

S	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	WVR8Q	

7.3.2.9.4

[GUIDELINE] UPnP devices should complete the transmission of a SOAP response within 29 seconds.

[ATTRIBUTES]

S	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	UY7PT	

7.3.2.9.5

[GUIDELINE] A UPnP control point may terminate the TCP connection for a SOAP response transmission that exceeds 30 seconds.

[ATTRIBUTES]

O	C	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	QTXQ7	A

7.3.2.10 DDC UPnP Device Description Rules

7.3.2.10.1

[GUIDELINE] The total byte size of a device description file shall not exceed 20,480 bytes (20 KB). This byte limit includes the HTTP headers.

[ATTRIBUTES]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	DXIDH	

NOTE: Provides a known maximum size for device description documents.

7.3.2.10.2

[GUIDELINE] DLNA UPnP devices shall employ the <dlna:X_DLNADOC> XML element inside the <device> element of the device description document to indicate adherence to a particular DLNA Home Networked Device Interoperability Guidelines document version. The value of this element is the DLNA Device Class or DLNA Device Capability, a dash character, followed by the numeric version value of the Interoperability Guidelines document.

The <dlna:X_DLNADOC> element indicates DLNA compliance for a specific <device>, excluding its embedded devices listed in <deviceList>.

The value of the <dlna:X_DLNADOC> element is a string as defined below. Linear white spaces (LWS) are not implied in this definition below.

- dlnadoc-value = dlna-dev-class | [dlna-dev-capability “/“ capability-host] "-" dlna-version
- dlna-dev-class = "DMS" | "DMR" | "DMPr" | "M-DMS" | other-dev-class
- other-dev-class = *<"A" - "Z", "a" - "z", "-">
- dlna-dev-capability = "+RUISINK+" | "+RUISRC+" | other-dev-capability
- other-dev-capability = *<"A" - "Z", "a" - "z", "+">
- capability-host = "DMS" | "DMR" | "DMPr" | "M-DMS" | "DMP" | "M-DMP" | "DMC" | "M-DMC" | "M-DMD" | "M-DMU" | "MIU" | "M-NCF"
- dlna-version = major-version "." minor-version
- major-version = DIGIT

- minor-version = DIGIT DIGIT

The dlna-dev-class represents a Device Class of a DLNA device.

The dlna-dev-capability represents a (discoverable) Device Capability.

The capability-host represents a DLNA Device Class that hosts a DLNA Device Capability.

The dlna-version represents a version of Interoperability Guidelines supported by the DLNA Device Class.

An example of <dlna:X_DLNADOC> element is shown as follows:

```
<dlna:X_DLNADOC xmlns:dlna="urn:schemas-dlna-org:device-1-0">
    DMS-1.50
</dlna:X_DLNADOC>
```

[ATTRIBUTES]

M	A	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	8CA7M	

NOTE: Provides an easy way of distinguishing UPnP devices that are claimed as being DLNA Device Classes or DLNA Device Capabilities.

This guideline specifies the scoping rules for the <dlna:X_DLNADOC> element. Essentially, UPnP devices (in a device hierarchy) needs to be marked explicitly as being DLNA devices. Although the subject matter is technically out of scope, this guideline permits a non DLNA UPnP device to be listed in a device hierarchy that has DLNA devices.

The<dlna:X_DLNADOC> element can appear multiple times such as the case for a DMS and M-DMS combination device.

The value of capability_host can be a discoverable DLNA Device Class as well as a non-discoverable DLNA Device Class, since also non-discoverable DLNA Device Classes can host a discoverable DLNA Device Capability.

7.3.2.10.3

[GUIDELINE] The<dlna:X_DLNADOC> element may appear multiple times.

[ATTRIBUTES]

O	A	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	9YPQX	

7.3.2.10.4

[GUIDELINE] UPnP control points shall be matched against multiple <dlna:X_DLNADOC> elements. Specifically, a control point that claims to discover a particular type DLNA device class or device capability shall be able to discover that type of DLNA device class or device capability, even if the specific <dlna:X_DLNADOC> element of interest is not the first <dlna:X_DLNADOC> in the device description document.

[ATTRIBUTES]

M	A	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	8RDTJ	A

7.3.2.10.5

[GUIDELINE] The namespace "urn:schemas-dlna-org:device-1-0" shall be specified in the <root> element or the <dlna:X_DLNADOC> element and the namespace prefix shall be "dlna:".

[ATTRIBUTES]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	RDTJ6	

7.3.2.10.6

[GUIDELINE] UPnP control points shall ignore the element value of <dlna:X_DLNADOC>. For example, DLNA control points shall not filter out a DLNA device because the version value is different from expected.

[ATTRIBUTES]

M	A	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	YPQX5	A

NOTE: In the near-term, the <dlna:X_DLNADOC> version number is useful for testing purposes. Future guidelines will specify behavior for interoperability between newer and older DLNA devices and the purpose of this field might change.

7.3.2.10.7

[GUIDELINE] If a vendor builds an implementation of a Device Class (with zero or more Device Capabilities or Device Options) or a discoverable Device Capability, then the implementation shall comply with all mandatory portions of the Interoperability Guidelines for the specified dlna-version token 7.3.2.10.2. If a vendor implements an older version of the Interoperability Guidelines, then the vendor shall not implement guidelines defined in newer versions of Interoperability Guidelines with the following exception:

- Devices that implement older versions of the Interoperability Guidelines may support Media Format Profiles defined in newer versions.

[ATTRIBUTES]

M	C	HND +RUISINK+ +RUISRC+	MHD	MIU	[47]	CA7MF	

NOTE: Vendors are not permitted to selectively implement portions of newer versions of DLNA's Interoperability Guidelines.

For example, if a vendor wants to build a DMS that supports RTP in the DLNA-defined manner, then the vendor will implement the DMS according to the 1.50 (or newer) version of the Interoperability Guidelines, which includes using a value of "1.50" for the dlna-version token. A DMS implementation that uses a "1.00" value for the dlna-version but implements guidelines that are specific to newer versions of Interoperability Guidelines demonstrates a violation of this guideline.

The primary reason for this guideline is to ensure that newer implementations participate in the DLNA networking ecosystem in a manner that is consistent with the assumptions of those guidelines. Newer versions of Interoperability Guidelines are drafted with compatibility for previous versions, but newer versions sometimes have new mandatory requirements. The new mandatory guidelines often ensure the robustness of the network, which is needed when Interoperability Guidelines increase network complexity through new usages.

This guideline makes no claim on policy decisions about granting a DLNA logo/certification to implementations that use older versions of Interoperability Guidelines. DLNA reserves the right to require a baseline version of Interoperability Guidelines for future implementations.

7.3.2.11 DDC UPnP Embedded Device Support

7.3.2.11.1

[GUIDELINE] DLNA UPnP devices shall not have more than 6 total UPnP devices in the device hierarchy with a maximum depth of 4. Root devices have a depth of 1.

[ATTRIBUTES]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	XIDHZ	

NOTE: A UPnP control point will handle DLNA devices that include a combination or an aggregate of devices and services. A specific limit sets a bound on memory and processing requirements for control points. Note that the +RUISRC+ and +RUISINK+ device capabilities as defined for remote UI also account for a device in the device hierarchy.

7.3.2.11.2

[GUIDELINE] UPnP control points shall support device hierarchies that have up to a total of 6 DLNA devices with a maximum depth of 4. Root devices have a depth of 1.

[ATTRIBUTES]

M	L	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	QI57Q	A

7.3.2.11.3

[GUIDELINE] DLNA UPnP devices shall be *functionally independent* even if they are in the same device hierarchy. In other words,

- A DLNA UPnP device is identified as a <device> with the <dlna:X_DLNADOC> element.
- A DLNA UPnP device has no *functional dependency* with other UPnP devices in the device hierarchy.
- A DLNA UPnP device has no *functional dependency* with other DLNA UPnP devices in the device hierarchy.

A DLNA UPnP device (Device-A) is *functionally independent* if it does not require a control point to invoke a UPnP action of another UPnP device (Device-B) in order to put Device-A in a state for use with a DLNA compliant UPnP control point. Also note that the definition assumes Device-A and Device-B are in the same device hierarchy. Furthermore, Device-B might or might not be a DLNA device, as indicated by the presence of the <dlna:X_DLNADOC> element.

[ATTRIBUTES]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	TXQ7Y	

NOTE: These guidelines simplify control point implementations by not requiring them to know about any functional dependencies between DLNA UPnP devices found in a device hierarchy.

Although the subject matter is technically out of scope, this guideline does not prohibit the use of UPnP device that has functional dependence on another UPnP device. However, a device that has a functional dependence cannot be marked with a <dlna:X_DLNADOC> element.

7.3.2.11.4

[GUIDELINE] DLNA control points shall not assume any *functional dependency* between embedded devices that contain the <dlna:X_DLNADOC> element.

For example, a control point that requires UPnP actions to be called on one of the UPnP devices before calling UPnP actions on another UPnP device (both UPnP devices belong to the same hierarchy and have the <dlna:X_DLNADOC> element) is in violation of this requirement.

[ATTRIBUTES]

M	L	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	Y7PTR	A

7.3.2.11.5

[GUIDELINE] UPnP devices may be implemented as a descendent of a UPnP root device, which might or might not be a standard UPnP device type.

[ATTRIBUTES]

O	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	VR8QA	

7.3.2.11.6

[GUIDELINE] UPnP control points shall interoperate with embedded DLNA devices that exist in device hierarchies where the root happens to be a non-standard UPnP device type (i.e. vendor-defined UPnP device type).

[ATTRIBUTES]

M	L	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	T6RCX	A

7.3.2.11.7

[GUIDELINE] UPnP devices that are not DLNA-compliant may be listed in a device hierarchy.

These non-DLNA UPnP devices count against the maximum number of 6 total UPnP devices in the device hierarchy, indicated in 7.3.2.11.1.

[ATTRIBUTES]

O	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	WQPE7	

NOTE: This guideline permits that a device hierarchy to have UPnP devices that do not have the <dlna:X_DLNADOC> element

7.3.2.12 DDC UPnP Service Description Rules

7.3.2.12.1

[GUIDELINE] Optional actions listed in the SCPD shall be supported and not return the NOT_IMPLEMENTED UPnP error in response to an invocation. Optional actions that are not implemented shall not be listed in the SCPD.

[ATTRIBUTES]

M	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	YZ8ZI

NOTE: UPnP devices will fully and accurately reflect capabilities required by the standardized Device Control Protocol (DCP) and listed in the UPnP device's service control protocol document (SCPD).

7.3.2.12.2

[GUIDELINE] A UPnP state variable shall not be present unless it meets at least one of the following characteristics.

- The UPnP state variable is actually used by the device, either as an evented state variable or as an action parameter.
- The UPnP state variable is normatively defined by a UPnP service to neither be evented nor used for an action parameter.

[ATTRIBUTES]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	LGQXH

NOTE: This guideline is in consideration of the fact that control points can run on a platform with limited resources.

7.3.2.12.3

[GUIDELINE] If an allowed value list or value range is specified, UPnP devices should accept all values in the state variable range, regardless of the stepping (as indicated by a <step> element of the UPnP state variable).

[ATTRIBUTES]

S	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	GQXHV

NOTE: Although it is preferable for control points to employ logic for correctly checking an argument for compliance against a device's stepping, this is not always the case. For broader interoperability, this guideline is suggested for UPnP devices but it is not mandatory. Note that the AVTransport, ContentDirectory, and ConnectionManager services do not have state variables that use stepping by default.

7.3.2.12.4

[GUIDELINE] Services with evented state variables shall support SUBSCRIBE and UNSUBSCRIBE operations.

[ATTRIBUTES]

M	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	Z8ZIU

NOTE: Specifies normative behavior for services with evented state variables.

7.3.2.12.5

[Guideline] DCP-required or SCPD-specified state variables with the attribute SendEvent="Yes" shall actually be evented.

[Attributes]

M	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	QPE7G

7.3.2.12.6

[Guideline] Service description files shall not exceed 51,200 bytes (50 KB). This byte limit includes the HTTP headers.

[Attributes]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	6RCXP

NOTE: This provides a reasonable maximum length for service description files.

7.3.2.13 DDC UPnP XML Namespace

[Guideline] Default namespace defined by the UPnP DA shall be used in device and service descriptions.

[Attributes]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	R8QAV

NOTE: All standard elements in device and service descriptions do not use a namespace prefix.

7.3.2.14 DDC UPnP Action Argument Encoding

7.3.2.14.1

[Guideline] The number, names and ordering of arguments of SOAP actions in an SCPD shall be identical to what is specified in the standardized DCP.

[Attributes]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	7PTRW

7.3.2.14.2

[Guideline] A UPnP device shall parse and interpret a SOAP action request in which the number, names and ordering of input arguments are identical to what is specified in the corresponding SCPD (which is the same as specified in the standardized DCP).

[Attributes]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	RW8QR

7.3.2.14.3

[Guideline] A UPnP device response to a SOAP action shall contain the number, name, and ordering of output arguments that are identical to what is specified in the corresponding SCPD (which is the same as specified in the standardized DCP).

[Attributes]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	9TZAV	

7.3.2.15 DDC UPnP SOAP Packet Size

7.3.2.15.1

[Guideline] UPnP devices shall be able to accept SOAP requests that are up to 20,480 bytes (20 KB) in size. This byte limit includes the HTTP headers.

[Attributes]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	XQ7YM	

NOTE: This guideline provides control points with a minimal SOAP packet size (total size for headers and body). It is understood the support of larger SOAP requests is permitted.

7.3.2.15.2

[Guideline] UPnP control points shall be able to accept SOAP responses that are up to 204,800 bytes (200 KB) in size. This byte limit includes the HTTP headers.

[Attributes]

M	L	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	I57Q8	A

NOTE: Security recommendations call out 200 KB as a reasonable upper bound for SOAP responses (total size for headers and body).

7.3.2.15.3

[Guideline] UPnP control points may refuse SOAP responses that are more than 204,800 bytes (200 KB) in size. This byte limit includes the HTTP headers.

Control points may implement the not accept SOAP response behavior by terminating the TCP connection after 200 KB is reached.

[Attributes]

O	L	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	IDHZS	A

7.3.2.16 DDC UPnP Error Codes

7.3.2.16.1

[GUIDELINE] Unless otherwise specified, UPnP endpoints (devices and control points) should use and return the proper error code when encountering an error condition for a UPnP operation. This includes using the proper HTTP error codes and method error codes for UPnP actions. In some extreme circumstances, it might be necessary to simply close a UPnP initiated connection upon encountering an error condition.

[ATTRIBUTES]

S	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[47]	A7MFW	A

NOTE: This requirement covers the proper expected behavior for any UPnP endpoint and is repeated here due to its importance in gracefully recovering from error conditions on a distributed home network.

7.3.2.16.2

[GUIDELINE] UPnP control points shall be able to tolerate unknown method error codes for UPnP actions

[ATTRIBUTES]

M	A	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	PQX5U	A

NOTE: Ideally, UPnP control points treat all unknown method error codes for UPnP actions as a generic error condition.

7.3.2.16.3

[GUIDELINE] HTTP clients for UPnP endpoints (devices and control points) are not required to understand unknown HTTP status code values, but they shall understand the class of the status code. The class of the status code is indicated by the first digit of the status code numeric value. HTTP clients shall treat unrecognized status code values as equivalent to the x00 code of the class.

[ATTRIBUTES]

M	C	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	DTJ6V	A

7.3.2.16.4

[GUIDELINE] UPnP devices should not use the UPnP error code value of 402 when a received SOAP action request contains arguments with unknown argument names.

[ATTRIBUTES]

S	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	WC47X	

NOTE: The proper way of handling a SOAP action request that contains arguments with unknown argument names is to ignore those arguments (7.3.2.23) and process the request as if those arguments have never existed.

The UPnP DA 1.0 requires devices to ignore unknown elements in all XML fragments, including unknown arguments in SOAP requests. However, the UPnP DA 1.0 also allows the use of the UPnP error code value of 402 (Invalid Args) to indicate that unknown arguments are found in SOAP requests. This error in the specification has since been acknowledged and corrected in the draft version of the UPnP DA v1.1. As a result of this ambiguity, certain existing devices issue the UPnP error code value of 402 (Invalid Args) in the event of unknown arguments. This behavior is discouraged but it does enable backward compatibility.

7.3.2.17 DDC UPnP GENA Packet Size

7.3.2.17.1

[GUIDELINE] UPnP control points shall be able to accept GENA event transmissions that are up to 20,480 bytes (20 KB) in size. This byte limit includes the HTTP headers.

[ATTRIBUTES]

M	L	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	TJ6VJ	A

NOTE: This guideline specifies the minimum capability of control points to receive events of 20 KB in size (for headers and body). Control points are permitted to support larger GENA events.

7.3.2.17.2

[GUIDELINE] UPnP control points may choose not to accept GENA event transmissions that are more than 20 KB in size.

Control points may implement the not accept GENA event behavior by terminating the TCP connection after 20 KB is reached.

[ATTRIBUTES]

O	L	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	QX5U5	A

7.3.2.18 DDC UPnP Subscription Handling

7.3.2.18.1

[GUIDELINE] The SUBSCRIBE response shall include the Content-Length: 0 HTTP header/value pair, if the response is not encoded with *Chunked Transfer Coding*.

The only exception to this rule is if the device can guarantee a TCP 'FIN' packet is sent before the initial event message is sent to the subscribing control point.

[ATTRIBUTES]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	7MFWJ

NOTE: In order for a control point to receive the initial event from a UPnP device, a control point needs to know the Subscription ID (SID) value.

The SID is obtained in the response to a SUBSCRIBE request.

Therefore, a control point will receive the entire SUBSCRIBE response before it receives the first event.

The HTTP clients of control points only have two ways to know when the SUBSCRIBE response has finished. The first is to complete the transaction when the Content-Length:0 values are specified. The second is to receive the TCP 'FIN' flag in the TCP stream.

7.3.2.18.2

[GUIDELINE] UPnP devices shall assign a globally unique SID, where the global context is defined as the UPnP network. The format for the uuid is as specified in 7.3.2.19.

[ATTRIBUTES]

M	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	DHZSM

NOTE: See 7.3.2.20 for a way to generate a globally unique SID.

7.3.2.19 DDC UPnP UUID Format

[GUIDELINE] The format of the SID is "uuid:" followed by a UUID, which is a 128-bit value represented in hexadecimal form, with optional hyphens throughout the encoding. The maximum length is 68 bytes, including the "uuid:" portion.

Example:
uuid:00000000-0000-0000-0000-000000000000

[ATTRIBUTES]

M	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	57Q8X

7.3.2.20 DDC UPnP UUID Generation

[GUIDELINE] UPnP devices should use the DCE 1.1 methodology for generating a globally unique UUID value.

[ATTRIBUTES]

S	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[75]	Q7YMT

NOTE: There are several ways to generate a UUID value. The best ways to generate a UUID involve using some form of a network address and the current time, such as the algorithm described in [75].

7.3.2.21 DDC UPnP Event Subscription Renewals

7.3.2.21.1

[General] This guideline instructs developers that the control point is responsible for renewing subscriptions in a timely manner.

7.3.2.21.2

[Guideline] If UPnP control points want to continue receiving UPnP events, then they shall renew their subscriptions before the negotiated subscription TIMEOUT expires.

[Attributes]

M	C	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	PTRWR

7.3.2.22 DDC UPnP Event Notification Handling

7.3.2.22.1

[Guideline] UPnP devices shall send events to all properly subscribed UPnP control points. The device shall enforce a subscription TIMEOUT value of 5 minutes.

The UPnP device behavior of enforcing this 5 minutes TIMEOUT value is implemented by specifying "TIMEOUT: second-300" as an HTTP header/value pair.

[Attributes]

M	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	8QAVJ

NOTE: A UPnP control point that subscribes to events and then subsequently leaves the UPnP network will cause a UPnP device to possess an event subscription to an invalid address. The device will not hold up events to other subscribing control points while, for example, the HTTP session with the absent UPnP control point times out. This scenario has been a major cause of UPnP control point disruption and usability problems. UPnP control points that stop receiving events might incorrectly indicate to the user that a device is stalled or is malfunctioning.

7.3.2.22.2

[Guideline] UPnP devices should monitor their subscription lists and remove control points that fail to renew their subscription within the negotiated time.

[Attributes]

S	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	RCXPQ

7.3.2.23 DDC UPnP Unknown Header/Tag/Field Robustness Rule

[Guideline] UPnP endpoints (devices and control points) shall be tolerant of unknown headers, tags, fields, attributes, and values for HTTP, SSDP, XML, SOAP, and GENA. Specifically, this tolerance guideline applies to:

- HTTP headers, tokens, values
- SSDP headers, tokens, values

- Unknown XML elements and attributes of SOAP or GENA fragments
- Unknown XML elements and attributes in device description files or service description files

Tolerant behavior is defined as being able to successfully "parse and interpret" or "parse and ignore" the unknown text.

[ATTRIBUTES]

M	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[21] [33] [47]	PE7GO	A

NOTE: This guideline addresses forward compatibility and also ensures broader interoperability between implementations that employ vendor extensions in the manner described by the guideline.

7.3.2.24 DDC URI Rules

7.3.2.24.1

[GUIDELINE] All absolute URIs used for *UPnP communications* shall use IP addresses (not host names).

UPnP communications specifically refers to the following areas.

- SOAP actions
- GENA events
- Device description files
- Service description (SCPD) files
- UPnP presentation files
- SSDP messages

[ATTRIBUTES]

M	L	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[47]	8ZIUM	A

NOTE: These guidelines are mandatory because DLNA Device Classes cannot depend on DNS infrastructure within a home network environment.

7.3.2.24.2

[GUIDELINE] The a.b.c.d format for Ipv4 addresses shall be used for *UPnP communications*, where each quad represents a byte in network byte order form.

[ATTRIBUTES]

M	L	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[47]	QXHV3	A

7.3.2.24.3

[GUIDELINE] HTTP URI escaping is always performed according to the URI specification ([19]) as required in subclause 3.2.1 of the HTTP/1.1 specification ([33]).

[ATTRIBUTES]

M	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[19] [33]	XHV3H	A

NOTE: This guideline specifies how to escape URI values.

7.3.2.24.4

[GUIDELINE] All URIs used for *UPnP communications* shall not exceed 256 bytes in URI-escaped UTF-8 encoded form. This guideline applies to both absolute URIs and complete URIs (relative URIs combined with a base path).

UPnP communications does not cover informational URIs for the manufacturer or product/model used inside UPnP Device Descriptions, and does not cover the URIs as defined in the UPnP Device Extension Schema's as defined in Annex A and Annex B of [101] for RUI Source and RUI Sink capabilities. It also does not cover indirectly referenced content, such as URIs inside the presentation files.

[ATTRIBUTES]

M	L	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[47]	ZIUM5	A

NOTE: These guidelines provide a maximum URI length for the UPnP layer. See guideline 7.4.1.3.10 sub-guideline 7.4.1.3.10.4 for the maximum URI length at the UPnP AV layer.

According to subclause 2.10 of [55], white spaces are significant (i.e. non-markup characters) in XML elements that contain character data. Therefore XML elements that contain a single (absolute or relative) URI value cannot have preceding or trailing white spaces.

7.3.2.24.5

[GUIDELINE] All URIs (not used for *UPnP communications*) shall not exceed 1024 bytes, in the URI-escaped UTF-8 encoded form. This guideline covers URIs, such as (but not limited to):

- URIs inside UPnP presentation files
- URIs in the device description for product, model, or manufacturer informatioN

- URIs as defined in the UPnP Device Extension Schema's as defined in Annex A and Annex B of [101] for RUI Source and RUI Sink capabilities.
- URIs used inside the XML UI Listing as defined in subclause 5.1.1.5 of [101] for RUI Source capabilities.
- URIs used inside CE-HTML pages as defined in [101].

[ATTRIBUTES]

M	L	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[47]	E7GO8	A

7.3.2.24.6

[GUIDELINE] UPnP devices should not use the <URLBase> element in the device description document.

[ATTRIBUTES]

S	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[29] [47]	CXPQK	

NOTE: These requirements have several benefits. Since the device description and service description documents will no longer include IP addresses and port numbers, UPnP devices are simplified. The document can be sent even if the IP address changes or the device is multi homed. UPnP control points will have an easier time dealing with UPnP devices that meet these requirements, and will be able to handle any situation that arises.

The terms Base URI, Relative URI, and Absolute URI are used here in a manner consistent with their definitions introduced in [29].

7.3.2.24.7

[GUIDELINE] If a URI in a device description is used for SOAP actions, GENA events, SCPD files, or UPnP presentation files, then the URI may be a Relative URI as defined in [29] with its Base URI as defined in Guideline 7.3.2.24.11.

[ATTRIBUTES]

O	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[29] [47]	QAVJB	

7.3.2.24.8

[GUIDELINE] UPnP control points shall work with UPnP devices that use a <URLBase> element and with those that do not use a <URLBase> element. Control points shall also work with UPnP devices that use Absolute or Relative URIs.

[ATTRIBUTES]

M	R	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU M-DMD	MIU	[29] [47]	TRWRA	A

7.3.2.24.9

[GUIDELINE] UPnP devices shall use the CALLBACK URI value sent by control points for event delivery, provided that the CALLBACK URI value is consistent with guidelines 7.3.2.24.1 - 7.3.2.24.4.

[ATTRIBUTES]

M	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	7YMTP	

7.3.2.24.10

[GUIDELINE] UPnP control points shall not specify more than one CALLBACK URI value for the CALLBACK header in a request with the SUBSCRIBE method.

[ATTRIBUTES]

M	L	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU M-DMD	MIU	[47]	7Q8XF	A

NOTE: Simplifies a UPnP device's implementation for GENA eventing.

7.3.2.24.11

[GUIDELINE] If the <URLBase> element is used, it shall define the Base URI to be used for declaring Relative URIs. When this element is omitted, the LOCATION value (URL) in the device advertisement shall define the Base URI.

[ATTRIBUTES]

M	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS		[29] [47]	XZ95C	

7.3.2.24.12

[GUIDELINE] Relative URI and a Base URI shall be resolved into an Absolute URI according to the process defined in [29].

Examples.

- The ssdp:alive message has LOCATION value of http://172.16.0.2/MyDir/devicedesc.xml
- The device description file does not have the <URLBase> element.
- One of the services has these element values.
  - <SCPDURL> has "/service_desc.xml"
  - <controlURL> has "control"
  - <eventSubURL> has "http://172.16.0.2:3000/sub"
- The Absolute URL for that service is as follows:
  - SCPDURL: http://172.16.0.2/service_desc.xml
  - controlURL: http://172.16.0.2/MyDir/control
  - eventSubURL: http://172.16.0.2:3000/sub

[ATTRIBUTES]

M	R	DMP +RUIPL+ +RUICTRL+ +SR+ +EPG+	n/a		[29] [47]	W8QR3	A

NOTE: In terms of syntax, reference [29] defines an Absolute URI and a Relative URI. A Relative URI is further classified as belonging to one of three types:

- Network path (net_path)
- Absolute path (abs_path)
- Relative path (rel_path)

These 3 types of Relative URIs can be resolved into Absolute URIs following the procedures defined in [29].

In the example introduced in Guideline 7.3.2.24.12, the SCPDURL value is an example of a Relative URI with an Absolute path. The controlURL value is an example of a Relative URI with a Relative path. The eventSubURL value is an example of an Absolute URI.

7.3.2.25 DDC UPnP Device description Usage

7.3.2.25.1

[GUIDELINE] If a DLNA UPnP device wants to change a device description or a service description files, then the UPnP device shall need to

- first leave the UPnP network by sending an ssdp:byebye message,
- then change the desired device description or service description files, and
- finally join the UPnP network with the new XML files using an ssdp:alive message.

[ATTRIBUTES]

M	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	HZSMF	

NOTE: UPnP control points often bind UDN values to device representations. Therefore, a device that changes its logical representation causes problems if it uses the same UDN. This guideline does not apply if the device sends an ssdp:byebye message. In such cases, the device can still keep its UDN value and change its logical representation before rejoining the UPnP network.

7.3.2.25.2

[GUIDELINE] A DLNA UPnP control point that removes a UPnP device from its list of active devices shall also invalidate its local representation of the device.

The control point removes a device for a variety of reasons, such as a CACHE-CONTROL timeout.

Invalidating the local representation of the device means that the control point shall reload the device description and service description files.

[ATTRIBUTES]

M	C	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	MFWJU	A

NOTE: This guideline obligates a control point to refresh device description and service description documents the next time the device is discovered. This, for examples, allows a device to add additional supported actions or services (via firmware update), without having to change the UDN of the device.

7.3.2.26 DDC UPnP UDN Usage

7.3.2.26.1

[GUIDELINE] UPnP devices should not change the UDN between reboots or application launch/shutdown.

[ATTRIBUTES]

S	A	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	X5U5G

NOTE: Implementing this guideline enables usages such as my favorite devices. Preferably, UPnP device UDN values will be long-lived.

UPnP DA states as follows;

UDN: Must be the same over time for a specific device instance (i.e., must survive reboots).

7.3.2.26.2

[GUIDELINE] UPnP devices shall not change the UDN if only the <friendlyName> or IP addresses values are changed.

[ATTRIBUTES]

M	A	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	J6VJQ

NOTE: UPnP control point can identify UPnP devices even if FriendlyName or IP addresses are changed.

7.3.2.26.3

[GUIDELINE] In conjunction with the restrictions in 7.3.2.26.2, UDN may be changed if a UPnP device changes its device description or any of its supported services.

[ATTRIBUTES]

O	A	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	6VJQP

7.3.2.26.4

[GUIDELINE] If a UPnP device UDN changes, it shall re-advertise on the network using the new UDN.

[ATTRIBUTES]

M	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	5U5GU

NOTE: Control points that receive the advertisement know that a new UPnP device is available. This is required by guideline 7.3.2.10.

7.3.2.26.5

[GUIDELINE] If a UPnP device UDN changes, it shall send an ssdp:byebye for the old UDN.

[ATTRIBUTES]

M	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	FWJUQ

NOTE: Without this guideline, UPnP control points will have no idea that the old UDN is no longer valid. This is required by guideline 7.3.2.10.

7.3.2.26.6

[GUIDELINE] A UPnP device shall limit their UDN to a UTF-8 encoded string value containing "uuid:" followed by a UUID as specified in 7.3.2.19.

[ATTRIBUTES]

M	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	ZSMFB

NOTE: See 7.3.2.20 for a way to generate a globally unique 128-bit value for the UDN.

7.3.2.27 DDC UPnP Multi Homing Rules

7.3.2.27.1

[GENERAL] Multiple home network segments, wireless networking, and Auto IP can combine to create usability problems that can be avoided by following the specified rules.

7.3.2.27.2

[GUIDELINE] When a UPnP device has multiple IP addresses, the device may advertise on those IP addresses with the same or different UDN.

[ATTRIBUTES]

O	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	Q8XFC

NOTE: Multiple home network segments, wireless networking, and Auto IP can combine to create usability problems that can be avoided by following the specified rules.

7.3.2.27.3

[GUIDELINE] The LOCATION URL value in a UPnP device advertisement shall contain the return IP address of the home network interface it is sent on.

[ATTRIBUTES]

M	C	DMS DMR DMPr +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+	M-DMS	n/a	[47]	YMTPX

7.3.2.27.4

[GUIDELINE] Upon receiving multiple advertisements for the same UPnP device UDN, a UPnP control point should select the vendor-defined preferred advertisement as the route to the device.

[ATTRIBUTES]

S	C	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	RWRAB	A

7.3.2.27.5

[GUIDELINE] When a UPnP control point gets an advertisement for a UPnP device UDN on a different IP address from the one it has previously selected, it may continue to use its selected IP address provided that it has received an advertisement on the selected IP address in the last 10 seconds. Otherwise, if the UPnP control point does not receive an advertisement for its selected IP address in the next 10 seconds, it may change its selection to the new IP address. Even if the control point keeps the selected IP address in this case, it should change its selection to the new IP address when an access to the selected IP address fails.

[ATTRIBUTES]

O	L	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU, M-DMD	MIU	[47]	AVJBK	A

NOTE: The "selected IP address" is as specified in guideline 7.3.2.27.3 and 7.3.2.27.4.

7.3.2.28 DDC UPnP Device Icons

7.3.2.28.1

[GUIDELINE] If a UPnP device provides a device icon, the UPnP device shall provide two JPEG icons that conform to the 7.1.8 of [56] and 7.1.9 of [56] media format profiles and two PNG icons that conform to the 7.2.2 of [56] and 7.2.3 of [56] media format profiles.

[ATTRIBUTES]

M	L	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47] [56]	XPQKE	

NOTE: This requirement will ensure device icon compatibility and good authoring practices. The reason for requiring PNG icons is that the lossless compression is much better for small size images. Furthermore, alpha-blending makes it possible to present better user interfaces.

7.3.2.28.2

[GUIDELINE] UPnP devices may provide additional icons in other formats besides PNG and JPEG.

[ATTRIBUTES]

O	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	7GO8Q	

7.3.2.28.3

[GUIDELINE] UPnP devices shall use <mimetype>, <width>, <height>, <depth>, and <url> sub-elements for an <icon> element within its device description.

The value of <mimetype>, <width>, and <height> elements for DLNA device icons shall conform to the DLNA icon media format profiles 7.1.8 of [56], 7.1.9 of [56], 7.2.2 of [56] and 7.2.3 of [56]respectively.

Values in Table 14 are recommended for the <depth> element which indicates color bits per pixel for PNG and JPEG device icons.

Table 14 — Color Depth of Device Icons

Icon Image Data		**<depth> Values**
PNG	Grayscale: 8 bits	8
	Grayscale: 16 bits	16
	Truecolor: 24 bits (triplet of 8 bits R/G/B samples)	24
	Indexed - color bits 24 bits (palette entry is a triplet of 8 bits R/G/B samples)	24
	Grayscale w/ alpha: 8 bits (with matching alpha channel depth)	8
	Grayscale w/ alpha: 16 bits (with matching alpha channel depth)	16
	Truecolor w/ alpha: 24 bits (triplet of 8 bit R/G/B samples, alpha channel shall be 8 bits)	24
JPEG (8 bits Y/Cr/Cb samples)		24

[ATTRIBUTES]

M	R	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47] [56]	IUM5G

NOTE: UPnP defines the way to indicate profiles for icon images.

Since <depth> value is unclear for PNG grayscale/index colored /alpha blending and JPEG, this guideline encourages use of the values for <depth> element required by the UPnP DA. Note that the values for PNG do not help to identify color types.

7.3.2.29 DDC UPnP UTF-8 Support

7.3.2.29.1

[GUIDELINE] UPnP endpoints (devices and control points) shall use UTF-8 encoding of all XML fragments. UPnP endpoints shall be tolerant of the UTF-8 maximum of 4 bytes of Unicode character as required by XML processors.

[ATTRIBUTES]

M	L	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[26] [47] [55]	HV3HU	A

NOTE: Specifying UTF-8 as the encoding method for *UPnP communications* provides the right balance for supporting a wide variety of languages without necessarily requiring devices to support all languages.

Although UTF-8 has characters that are encoded in 6 bytes, W3C XML spec states that XML processors must accept any character in Unicode. This means XML parsers will decode up to 4 bytes of character. Specifically, see subclause 2.2 of [55] for more information. It calls out any Unicode character, excluding the surrogate blocks, FFFE, and FFFF.

7.3.2.30 DDC UPnP XML Comments

7.3.2.30.1

[GENERAL] XML comments normally have to be skipped by XML parsers. This guideline ensures that comments do not prevent interoperation.

7.3.2.30.2

[GUIDELINE] UPnP endpoints (devices and control points) shall never source XML with comments. This does not hold for CE-HTML content served by a RUI Source capability.

[ATTRIBUTES]

M	L	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[47]	V3HUN	A

7.3.2.30.3

[GUIDELINE] UPnP endpoints (devices and control points) may reject any XML provided with comments. This does not hold for interpreting CE-HTML content served by a RUI Source capability on a RUI Pull Controller or a RUI Sink capability.

[ATTRIBUTES]

O	C	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[47]	UM5G6	A

7.3.2.31 DDC UPnP Boolean Types

7.3.2.31.1

[GUIDELINE] UPnP endpoints (devices and control points) shall use "0" for false and "1" for true when using the UPnP Boolean type.

[ATTRIBUTES]

M	L	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[47]	GO8Q5	A

NOTE: This simplifies control point implementations and also reduces the size of some UPnP traffic.

7.3.2.32 DDC CP Versioning

7.3.2.32.1

[GUIDELINE] UPnP action requests (sent by a control point) shall include a DLNA-CP-version in a USER-AGENT HTTP header value.

[ATTRIBUTES]

M	A	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU M-DMD	MIU	[21] [33] [47]	PQKEA	A

NOTE: The HTTP specifications specify the format of the USER-AGENT HTTP header and header value.

Note that the USER-AGENT HTTP header is not used exclusively for DLNA information.

7.3.2.32.2

[GUIDELINE] The syntax of DLNA-CP-version is a subset of the 'product' token syntax (defined by HTTP) and is described below.

- DLNA-CP-version = "DLNADOC/" dlna-version

The dlna-version token is defined in guideline 7.3.2.10.2.

Example:

- USER-AGENT: DLNADOC/1.50
- USER-AGENT: UPnP/1.0 DLNADOC/1.50
- USER-AGENT: CERN-LineMode/2.15 libwww/2.17b3 DLNADOC/1.50 UPnP/1.0

[ATTRIBUTES]

M	A	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU M-DMD	MIU	[21] [33]	VJBKF	A

NOTE: The DLNA-CP-version token uses the syntax of a 'product' token (as defined in the HTTP specifications) to identify the DLNA guidelines version.

Note that space and separators can not be used in a token, and the USER-AGENT header field uses the token syntax. For example, "My DLNA Device / 1234" does not comply with the token syntax because spaces are used in the string that is supposed to follow the token syntax.

7.3.2.32.3

[GUIDELINE] UPnP devices shall be tolerant of UPnP action requests that specify a newer dlna-version in the DLNA-CP-version token.

Tolerance means that the UPnP device responds according to the version indicated by the device's <dlna:X_DLNADOC> value.

[ATTRIBUTES]

M	C	DMS DMR DMPr +RUISINK+ +RUISRC+	M-DMS	n/a	[47]	WRABY	

NOTE: This guideline is another clarification of the 7.3.2.23 in that it requires tolerance of unknown HTTP headers and values.

This guideline essentially requires a DLNA-compliant UPnP device to respond to newer control points using the DLNA-defined rules employed by the UPnP device.

7.3.2.33 DDC Absolute and Relative URI Requests

7.3.2.33.1

[GUIDELINE] The HTTP server of UPnP endpoints (devices and control points) shall accept an HTTP request that specifies an absolute or relative URI in the HTTP request.

[ATTRIBUTES]

M	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	MTPX7	A

NOTE: The HTTP specification indicates that this behavior is required.

Absolute URIs are permitted in HTTP requests.

7.3.2.33.2

[GUIDELINE] The HTTP client of UPnP endpoints (devices and control points) should specify a relative URI in the HTTP request.

[ATTRIBUTES]

S	R	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	8XFCS	A

7.3.2.34 DDC Maximum HTTP Header Size

7.3.2.34.1

[GUIDELINE] HTTP clients and servers of UPnP endpoints (devices and control points) shall generate and parse HTTP messages that have a total HTTP header size that is equal to or less than 4096 bytes (4 KB) in all HTTP requests and responses.

The total HTTP header size is the total number of bytes from the first byte in the start-line token and the last byte of the CRLF token, as used in the generic-message token defined in subclause 4.1 of [33], as quoted in the syntax below.

- generic-message = start-line *(message-header CRLF) CRLF [message-body]

[ATTRIBUTES]

M	L	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	[33]	SMFBW	A

NOTE: This provides a reasonable assumption as to how much memory is necessary is for all HTTP headers used in a single transaction related to UPnP communication.

7.3.2.35 DDC Device Capabilities

7.3.2.35.1

[GUIDELINE] The <dlna:X_DLNACAP> is a comma-separated list of Capability ID values that appears at most once for each <device> element in the device description document. The syntax of the <dlna:X_DLNACAP> value, dlnacap-value, is defined as follows;

- dlnacap-value = capID *("," capID)
- capID= *<"a"-"z", "A"-"Z", "0"-"9", "_","-">

The capID token shall always be a value defined by the DLNA guidelines and the length of the token shall not exceed 512 bytes.

The name space for the <dlna:X_DLNACAP> shall be "urn:schemas-dlna-org:device-1-0" and the namespace prefix shall be "dlna:".

Example:

- <dlna:X_DLNACAP xmlns:dlna="urn:schemas-dlna-org:device-1-0">av-upload,image-upload,audio-upload</dlna:X_DLNACAP>

[ATTRIBUTES]

M	A	DMS DMR DMPr	M-DMS	n/a	[47]	WJUQC	

NOTE: Other guidelines define the Capability ID values that are permitted in the <dlna:X_DLNACAP> element. The discovery of DLNA RUI Device Capabilities uses element <dlna:X_DLNADOC> defined in 7.3.2.10.2.

7.3.2.35.2

[GUIDELINE] UPnP control points shall be tolerant of unknown Capability ID values.

Tolerant behavior is defined as being able to successfully "parse and interpret" or "parse and ignore" the unknown text.

[ATTRIBUTES]

M	A	DMP DMC +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMP M-DMC M-DMU M-DMD	MIU	[47]	U5GUW	A

7.3.2.36 DDC DLNAQOS Support

[GUIDELINE] If DLNAQOS as defined in 7.2 is implemented, all UPnP Device and Control Point traffic shall be tagged with DLNAQOS_1, or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3), in accordance with Table 11.

[ATTRIBUTES]

M	A	HND +PR1+ +PR2+ +DN+ +PU+ +UP+ +RUISINK+ +RUISRC+ +RUIPL+ +RUICTRL+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	MHD	MIU	n/a	VJQP3	A

7.4 Media Management

7.4.1 AV Media Management

7.4.1.1 General

This subclause of the DLNA Home Networked Device Interoperability Guidelines covers the guidelines for implementing media management using the UPnP AV architecture.

DLNA Home Networked Device Interoperability Guidelines version 1.0 had requirements for metadata that is distributed on the home network. These guidelines are now updated with new language for new Device Classes and Capabilities that implement the new system usages of DLNA v1.5.

It is important to note that DIDL-Lite can be used in multiple contexts such as the following items.

- Some UPnP AV action request can carry DIDL-Lite metadata as input argument values (e.g. CDS:CreateObject and AVT:SetAVTransportURI requests).
- Some UPnP AV action responses can carry DIDL-Lite metadata as output argument values (e.g. CDS:CreateObject, AVT:GetMediaInfo, and AVT:GetPositionInfo request).
- Some UPnP AV events can carry DIDL-Lite metadata (e.g. AVT.LastChange and virtual instance state variables such as AVT.CurrentTrackMetaData).

The general rules for handling XML documents and fragments are specified in 6.7

AV Media Management Guidelines is organized into multiple subclauses.

- 7.4.1.2 specifies the UPnP AV components that are needed for a DLNA Device Class or a DLNA Device Capability. For example, this sub-table provides the guideline that specifies that a DMP shall implement a UPnP AV MediaServer control point.
- 7.4.1.3 specifies general UPnP AV requirements that are used by a variety of UPnP AV devices and control points. Guidelines that dictate rules for things like DIDL-Lite metadata, protocolInfo values, and DLNA-defined parameters for the 4th field of protocolInfo values are in this sub-table.

- 7.4.1.4 provides guidelines that are specific to UPnP AV MediaServer devices. Occasionally, a related guideline that specifies behavior for a UPnP AV MediaServer control point will also appear in this sub-table.
- 7.4.1.6 provides guidelines that are specific to UPnP AV MediaRenderer devices. Occasionally, a related guideline that specifies behavior for a UPnP AV MediaRenderer control point will also appear in this sub-table.
- 7.4.1.7 provides UPnP AV guidelines that are related to the Upload System Usage. Guidelines that specify behavior for CDS:CreateObject and CDS:DestroyObject transactions appear in this sub-table.

7.4.1.2 Device Classes and Device Capabilities Requirements

7.4.1.2.1 MM UPnP AV Compliance

7.4.1.2.1.1

[General] The following requirements scope which version of the UPnP AV specifications are required to implement one or more DLNA System Usages by DLNA Device Classes and Device Capabilities.

7.4.1.2.1.2

[Guideline] The DLNA Device Classes and Device Capabilities that implement one or more of the MSCP, MSD, MRCP, and MRD Device Functions, as defined in 5.3, shall be compliant to the appropriate version of UPnP AV specifications as defined in guidelines 7.4.1.2.1.3 through 7.4.1.2.1.7.

[Attributes]

M	R	DMS DMP DMR DMC +DN+ +UP+ +PU+ +PR2+ +DNSYNC+ +UPSYNC+ +SR+ +EPG+	M-DMS M-DMP M-DMC M-DMD M-DMU	MIU	[44] [45] [46] [48] [49] [51] [60] [61]	JQP33	A

7.4.1.2.1.3

[Guideline] The DLNA Device Classes and Device Capabilities that implement one or more of the MSCP, MSD, MRCP, and MRD Device Functions, as defined in 5.3, shall implement at a minimum all of the mandatory portions for the appropriate UPnP AVv1 specifications to implement the Device Functions of the following System Usages, as defined in 5.7, unless overridden by 7.4.1.2.1.6.

- 2-Box Pull
- 2-Box Push
- 3-Box
- 3-Box Printing
- Download
- Upload.

[Attributes]

M	R	DMS DMP DMR DMC +DN+ +UP+ +PU+ +PR2+	M-DMS M-DMP M-DMC M-DMD M-DMU	MIU	[44] [45] [46] [48] [49] [51]	8IAT3	

NOTE: UPnP AVv1 specifications are the baseline architecture for the specified DLNA System Usages.

7.4.1.2.1.4

[GUIDELINE] The DLNA Device Classes and Device Capabilities that implement one or more of the MSCP and MSD Device Functions, as defined in 5.3, shall implement at a minimum all of the mandatory portions for the appropriate UPnP AVv2 specifications to implement the Device Functions of the following System Usage, as defined in 5.7, unless overridden by 7.4.1.2.1.6:

- Scheduled Recording

[ATTRIBUTES]

M	R	DMS +SR+	M-DMS	n/a	[44] [45] [46] [48] [49] [51] [60] [61]	AT3QS	C

NOTE: UPnP AVv2 is the baseline architecture for the specified DLNA System Usage.

7.4.1.2.1.5

[GUIDELINE] The DLNA Device Classes and Device Capabilities that implement one or more of the MSCP and MSD Device Functions, as defined in 5.3, shall implement at a minimum all of the mandatory portions for the appropriate UPnP AVv3 specifications to implement the Device Functions of the following System Usages, as defined in 5.7:

- Upload Synchronization
- Download Synchronization
- EPG

[ATTRIBUTES]

M	R	DMS +UPSYNC+ +DNSYNC+ +EPG+	M-DMS	n/a	[44] [45] [46] [48] [49] [51] [60] [61]	N7DWU	C

NOTE: UPnP AVv3 is the baseline architecture for the specified DLNA System Usage.

7.4.1.2.1.6

[GUIDELINE] The DLNA Device Classes and Device Capabilities that implement the MSD Device Function, as defined in 5.3, to participate in more than one of the System Usages, as defined in 5.7, shall implement at a minimum all of the mandatory portions for the highest UPnP AV version baseline requirement as defined in guidelines 7.4.1.2.1.3 through 7.4.1.2.1.5.

[ATTRIBUTES]

M	R	DMS	M-DMS	MIU	[45] [46] [49] [60] [61]	4RS9F	A

NOTE: The baseline AV architecture for a UPnP AV MediaServer is the highest mandated baseline for all of the System Usages implemented on a UPnP AV MediaServer. For example, a DMS that implements the 2-Box Pull, 3-Box and the Upload Synchronization System Usages, will need to implement to the AVv3 specifications (i.e. cannot implement to a mixture of AV specification versions on a UPnP AV MediaServer).

7.4.1.2.1.7

[GUIDELINE] The DLNA Device Classes and Device Capabilities that implement one or more of the MSCP, MSD, MRCP, and MRD Device Functions, as defined in 5.3 and 5.7, may implement the mandatory portions of the UPnP AV specifications above its required baseline (i.e. higher version) for any of the DLNA System Usages.

[ATTRIBUTES]

O	R	DMS DMP DMR DMC +DN+ +UP+ +PU+ +PR2+ +UPSYNC+ +DNSYNC+ +SR+ +EPG+	M-DMS M-DMP M-DMC M-DMD M-DMU	MIU	[44] [45] [46] [48] [49] [51] [60] [61]	JZ3X6	A

NOTE: Any of the DLNA System Usages can be implemented using a version of the UPnP AV specification beyond its baseline. For example, devices implementing the 2-Box Pull System Usage needs to implement to AVv1 at a minimum. But they are allowed to implement to the AVv2, AVv3, and any future versions of the UPnP AV specifications.

7.4.1.2.2 MM DMP/M-DMP UPnP AV MediaServer Control Point Definition

[GUIDELINE] A DMP and M-DMP shall implement a UPnP AV MediaServer control point for browsing a ContentDirectory service on a DMS and M-DMS respectively.

[ATTRIBUTES]

M	R	DMP	M-DMP	n/a	[46] [49]	5GUWH	

NOTE: This guideline indicates that a DMP Device Class will use a UPnP control point that controls a UPnP AV MediaServer for browsing content.

7.4.1.2.3 MM M-DMD/Download Controller Definition

7.4.1.2.3.1

[GUIDELINE] A DLNA Device Class may implement the Download Controller Device Capability.

[ATTRIBUTES]

O	A	DMS DMP DMR DMC DMPr	M-DMS M-DMP M-DMC M-DMU	n/a	n/a	96QAD	N

7.4.1.2.3.2

[GUIDELINE] A Download Controller and M-DMD shall implement a UPnP AV MediaServer control point for browsing a ContentDirectory service on a DMS or M-DMS.

[ATTRIBUTES]

M	R	+DN+	M-DMD	n/a	[46] [49]	JUQCY	C

NOTE: This guideline indicates that a Download Controller Device Capability will use a UPnP control point that controls a UPnP AV MediaServer for browsing content.

7.4.1.2.4 MM M-DMU/Upload Controller Definition

[GUIDELINE] A Upload Controller and M-DMU shall implement a UPnP AV MediaServer control point for uploading content to a DMS and M-DMS respectively.

[ATTRIBUTES]

M	R	+UP+	M-DMU	n/a	n/a	MFBWX	A

NOTE: This guideline indicates that an Upload Controller Device Capability will use a UPnP control point that controls a UPnP AV MediaServer for sending content to the MediaServer.

See 7.4.1.7.3.2 and 7.4.1.7.3.3 for the functionality that needs to be implemented

7.4.1.2.5 MM DMR UPnP AV MediaRenderer Device Definition

7.4.1.2.5.1

[GUIDELINE] A DMR shall implement a UPnP AV MediaRenderer device that shall have one AVTransport service, one RenderingControl service, and one ConnectionManager service.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[48]	XFCS3	C

NOTE: DMR device will implement the baseline services for a UPnP AV MediaRenderer. This is in conjunction with the requirements for Rendering Endpoints described in 7.5.

7.4.1.2.5.2

[GUIDELINE] A UPnP AV MediaRenderer shall identify in the Device Description Document the AVTransport service, the RenderingControl service, and the ConnectionManager service using serviceType and serviceId elements with the following values:

Table 15 — DMR serviceType and serviceId Values

Service	Element	Value
ConnectionManager service	serviceType	urn:schemas-upnp-org:service:ConnectionManager:V
	serviceID	urn:upnp-org:serviceId:ConnectionManager
AVTransport service	serviceType	urn:schemas-upnp-org:service:AVTransport:V
	serviceID	urn:upnp-org:serviceId:AVTransport
RenderingControl service	serviceType	urn:schemas-upnp-org:service:RenderingControl:V
	serviceID	urn:upnp-org:serviceId:RenderingControl

where V is the version number of the service.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[47] [48]	N3DAH	N

NOTE: The serviceType and serviceId values uniquely identify the type of service. The table above lists the element values in compliance with the related UPnP specifications.

7.4.1.2.6 MM DMR AVTransport Rules

[GUIDELINE] A DMR shall support the mandatory actions and state variables for a AVTransport service.

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[44]	TPX7T	

NOTE: This guideline specifies the minimum requirements for a DMR's AVTransport service.

7.4.1.2.7 MM DMR ConnectionManager Rules

[GUIDELINE] A DMR shall support the mandatory actions and state variables for a ConnectionManager service.

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[45]	RABYC	

NOTE: This guideline specifies the minimum requirements for a DMR's ConnectionManager service.

7.4.1.2.8 MM DMR RenderingControl Rules

[GUIDELINE] A DMR shall support the mandatory actions and state variables for a RenderingControl service.

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[51]	JBKFB	

NOTE: This guideline specifies the minimum requirements for a DMR's RenderingControl service.

7.4.1.2.9 MM DMC/M-DMC UPnP AV MediaServer and AV MediaRenderer Control Point Definition

[GUIDELINE] A DMC and M-DMC shall implement a UPnP AV MediaServer control point and a UPnP AV MediaRenderer control point. The MediaServer control point interacts with the ContentDirectory service for browsing content. The MediaRenderer control point interacts with the AVTransport service and the ConnectionManager service to verify that the MediaRenderer can play the content and to start and stop the playback.

[ATTRIBUTES]

M	R	DMC	M-DMC	n/a	[44] [45] [46] [48] [49]	QKEAC	

NOTE: This guideline indicates that the DMC and M-DMC Device Classes will use a UPnP control point that controls a UPnP AV MediaServer and a UPnP AV MediaRenderer.

7.4.1.2.10 MM Push Controller Definition

7.4.1.2.10.1

[GUIDELINE] A DLNA Device Class may implement the Push Controller Device Capability.

[ATTRIBUTES]

O	A	DMS DMP DMR DMC DMPr	M-DMS M-DMP M-DMC M-DMD M-DMU	n/a	n/a	KK4QV	N

7.4.1.2.10.2

[GUIDELINE] A Push Controller shall implement a UPnP AV MediaRenderer control point that interacts with the AVTransport service and the ConnectionManager service to verify that the MediaRenderer can play the content and to start and stop the playback.

[ATTRIBUTES]

M	R	+PU+	n/a	n/a	[44][45][48]	O8Q5K	

NOTE: A Push Controller is a capability that controls a DMR and serves the content directly to the DMR. In addition to this guideline, a Push Controller needs to implement additional guidelines related to MediaRenderer control points and Content Source endpoints.

7.4.1.2.11 MM DMS/M-DMS UPnP AV MediaServer Device Definition

7.4.1.2.11.1

[GUIDELINE] A DMS and M-DMS shall implement a UPnP AV MediaServer device that shall have one ContentDirectory service and one ConnectionManager service.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[49]	M5G6Y	C

NOTE: DMS and M-DMS devices needs to implement the minimum baseline services for a UPnP AV MediaServer.

7.4.1.2.11.2

[GUIDELINE] This guideline no longer applies.

[ATTRIBUTES]

7.4.1.2.11.3

[GUIDELINE] A UPnP AV MediaServer shall identify in the Device Description Document the ContentDirectory service and the ConnectionManager service using serviceType and serviceId elements with the following values:

Table 16 — DMS/M-DMS serviceType and serviceId Values

Service	Element	Value
ContentDirectory service	serviceType	urn:schemas-upnp-org:service:ContentDirectory:V [a]
	serviceID	urn:upnp-org:serviceId:ContentDirectory
ConnectionManager service	serviceType	urn:schemas-upnp-org:service:ConnectionManager:V
	serviceID	urn:upnp-org:serviceId:ConnectionManager
[a] V is the version number of the service		

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[47] [49]	MDNOR	N

NOTE: These serviceType and serviceId values uniquely identify the type of service. The table above lists the element values in compliance with the related UPnP specifications.

7.4.1.2.11.4

[GUIDELINE] A DMS and M-DMS may have a ScheduledRecording service in the UPnP AV MediaServer device.

[ATTRIBUTES]

O	R	DMS	M-DMS	n/a	[49] [60]	LPXZY	N

NOTE: When a UPnP AV MediaServer Device contains a ScheduledRecording service it's indicating that it has implemented the DLNA ScheduledRecording Device Option as specified in 7.4.4.

7.4.1.2.11.5

[GUIDELINE] If a DMS or an M-DMS contains a ScheduledRecording service then it shall implement the DLNA ScheduledRecording Device Option as specified in 7.4.4.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[49] [60]	YR8Q6	N

7.4.1.2.12 MM DMS/M-DMS ContentDirectory Rules

7.4.1.2.12.1

[GUIDELINE] A DMS and M-DMS shall support the mandatory actions and state variables for a ContentDirectory service.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	HUNQQ	

7.4.1.2.13 MM DMS/M-DMS ConnectionManager Rules

7.4.1.2.13.1

[GUIDELINE] A DMS and M-DMS shall support the mandatory actions and state variables for a ConnectionManager service.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[45]	5G6YN	

NOTE: This guideline specifies the minimum requirements for a DMS's and M-DMS's ConnectionManager service.

7.4.1.2.14 MM MIU Definition

7.4.1.2.14.1

[GUIDELINE] For every virtual renderer that the MIU hosts, it shall implement the Media Management functionality of a DMR device as defined in 7.4.1.2.5 through 7.4.1.2.8.

[ATTRIBUTES]

M	R	n/a	n/a	MIU	[48]	8Q5KY	

NOTE: The MIU hosts virtual devices that control and extend the functionality of native servers and renderers in the network.

7.4.1.2.14.2

[GUIDELINE] For every virtual renderer that the MIU hosts, it shall implement a UPnP AV MediaRenderer control point for controlling the native DMR.

[ATTRIBUTES]

M	R	n/a	n/a	MIU	[44] [45] [48] [90]	KEACY	

7.4.1.2.14.3

[GUIDELINE] For every virtual server that the MIU hosts it shall implement the Media Management functionality of either a DMS or M-DMS device as defined in guidelines 7.4.1.2.11 through 7.4.1.2.13.

[ATTRIBUTES]

M	R	n/a	n/a	MIU	[45] [46] [49]	BKFB4	

7.4.1.2.14.4

[GUIDELINE] For every virtual server that the MIU hosts, it shall implement a UPnP AV MediaServer control point for controlling the native DMS or M-DMS device.

[ATTRIBUTES]

M	R	n/a	n/a	MIU	[44] [45] [46] [49]	ABYCT	

7.4.1.3 General UPnP AV Requirements

7.4.1.3.1 MM UPnP AV Control Point Tolerance of Unknown Property

[GUIDELINE] UPnP AV control points shall be tolerant of all properties that appear in a DIDL-Lite XML fragment. Properties are as defined in Appendix B of [46].

Tolerant behavior is defined as being able to successfully "parse and interpret" or "parse and ignore" the DIDL-Lite XML elements and attributes.

[ATTRIBUTES]

M	C	DMP DMC +DN+ +PU+ +UP+ +PR2+	M-DMP M-DMC M-DMU, M-DMD	MIU	[46]	PX7T9

NOTE: This guideline ensures that a UPnP AV MediaServer control point will continue to behave properly even if a UPnP AV MediaServer device improperly implements its support of the Filter argument for CDS:Browse and CDS:Search.

7.4.1.3.2 MM DIDL-Lite Restrictions

[GUIDELINE] UPnP AV endpoints (devices and control points) shall never source the following in DIDL-Lite documents or fragments:

- [CDATA] payloads
- <!...> XML comments

DIDL-Lite documents and fragments are always assumed to have a UTF-8 encoding. The XML for DIDL-Lite documents and fragments shall never contain XML comments. UPnP AV endpoints may reject any XML that is not encoded with these restrictions.

[ATTRIBUTES]

M	L	DMS DMR DMC +PU+ +UP+	M-DMS M-DMC M-DMU	n/a	[46]	FCS3V

NOTE: The flexibility of XML can cause problems for a number of XML parsers, especially those on resource-limited platforms. These requirements balance the needs of control points with the limitations of some devices.

The assumption of using only UTF-8 encoded DIDL-Lite documents and fragments enables wide language support without requiring control points to handle multiple encoding formats.

7.4.1.3.3 MM DIDL-Lite Max Metadata Length

7.4.1.3.3.1

[GUIDELINE] Unless specified in another DLNA guideline, element values and attribute values, appearing in DIDL-Lite documents or fragments, that are length-unlimited shall not exceed 1024 bytes each, in their XML-escaped form, encoded in UTF-8.

Length-unlimited data types are the data types with an unspecified maximum length in string form. These include string, URI, bin.hex, and base64 values.

[ATTRIBUTES]

M	L	DMS DMR DMC +PU+ +UP+	M-DMS M-DMC M-DMU	n/a	[46]	FBWX6

NOTE: This guideline puts a worst-case limit on all other metadata values found in a DIDL-Lite document or fragment. This allows for smaller limits to be specified, but that at this time this is a true maximum.

This guideline applies only to simple element values that are string, URI, bin.hex and base64 values. It does not apply to element values of other data types. It also does not apply to complex elements that contain sub-elements.

7.4.1.3.3.2

[GUIDELINE] In DIDL-Lite documents or fragments, element and attribute values that are length-limited shall not exceed their implied lengths.

Length-limited data types are the data types with an implied maximum length in string form. These include signed/unsigned integers, floating point numbers, Boolean values, etc. The following table defines the maximum byte length for these data types that are used by the CDS and UPnP device architecture..

Table 17 — CDS and UPnP Max Byte Length

Data Type in CDS	Data Type in UPnP Device Architecture	Maximum Byte Length
boolean	boolean	5
unsigned integer	ui4	10
integer	i4	11
unsigned long	n/a	20
long	n/a	21
n/a	ui1	3
n/a	ui2	5
n/a	i1	4
n/a	i2	6
n/a	int	11
n/a	r4	14
n/a	r8	22
n/a	number	22
n/a	fixed.14.4	20
n/a	float	110 [a]
n/a	char	1
n/a	date	10
n/a	dateTime	19
n/a	dateTime.tz	29
n/a	time	8
n/a	time.tz	18
[a] Float must be in the canonical representation		

[ATTRIBUTES]

M	L	DMS DMP DMR DMC +DN+ +PU+ +UP+ +PR2+	M-DMS M-DMP M-DMC M-DMD M-DMU	MIU	[46]	UQCYR

NOTE: A float value is m × 2^e, where m is an integer whose absolute value is less than 2^24, and e is an integer between -149 and 104, inclusive.

Lexical representation is as follows;

float-value := mantissa [("E" | "e") exponent] | "0" | "-0" | "INF" | "-INF" | "NaN"

mantissa := ["+" | "-"] 1*DIGIT ["." 1*DIGIT]

exponent := ["+" | "-"] 1*DIGIT

The canonical representation is defined; but in the non-canonical representation, the byte length can be infinite. In the canonical representation, the maximum byte length is 110.

For example, -1E4, 1267.43233E12, 12.78e-2, 12 and INF are all legal literals for float.

See: **http://www.w3.org/TR/xmlschema-2/#float**

A Boolean value can be either "0", "1", "true", or "false". The maximum length is set to 5 which is the size for the value "false". Even though guideline 7.3.2.31 restricts using Boolean values to "0" and "1" for DLNA Device Classes, it needs to be tolerant that "true" or "false" might be encountered for non DLNA devices. Hence the reason for setting the maximum length to 5.

7.4.1.3.3.3

[Guideline] The dc:title metadata property should not exceed 256 bytes in the XML-escaped form encoded in UTF-8.

[Attributes]

S	L	DMS DMP DMR DMC +DN+ +PU+ +UP+ +PR2+	M-DMS M-DMP M-DMC M-DMD M-DMU	n/a	[46]	GUWHV

NOTE: Although the maximum length for the dc:title is 1024 bytes, many titles can fit in 256 bytes. The primary reason why title values are allowed to exceed 256 bytes is to accommodate the guideline 7.4.1.4.18.

7.4.1.3.3.4

[Guideline] The following metadata properties shall not exceed 256 bytes each in the XML-escaped form encoded in UTF-8.

- upnp:class
- Any length-unlimited metadata property in Table 19. Recommended Metadata Properties.
- All length-unlimited DIDL-Lite schema defined attributes for <res>, except URI. (The length for URI is governed by guideline 7.4.1.3.10.) .

[Attributes]

M	L	DMS DMP DMR DMC +DN+ +PU+ +UP+ +PR2+	M-DMS M-DMP M-DMC M-DMD M-DMU	n/a	[46]	QP334

NOTE: This guideline provides devices and control points that receive metadata with some information about how much memory will be needed to represent a CDS object.

For example URIs include res@importUri, res@dlna:ifoFileURI, and res@dlna:importIfoFileURI.

7.4.1.3.4 MM DIDL-Lite Non-empty Metadata Values

7.4.1.3.4.1

[General] UPnP AV endpoints that provide DIDL-Lite metadata with empty values or values composed entirely of white-spaces can cause problems for other endpoints that receive them. More importantly, a user has little idea on how to interpret such values.

7.4.1.3.4.2

[Guideline] UPnP AV endpoints (devices and control points) shall use non-empty and non-whitespace values for *metadata properties* in a DIDL-Lite XML fragment. The term, *metadata* *properties*, refers specifically to elements and attributes defined by the DIDL-Lite schema and applies only to guideline entries of 7.4.1.3.4.

Exceptions are explicitly allowed in other guidelines, such as 7.4.1.7.19 and 7.4.1.7.23.

[ATTRIBUTES]

M	L	DMS DMR DMC +UP+	M-DMS M-DMC M-DMU	n/a	[46]	P334S	

7.4.1.3.4.3

[GUIDELINE] This guideline no longer applies.

[ATTRIBUTES]

		n/a	n/a	n/a	n/a	UWHVV	D

7.4.1.3.4.4

[GUIDELINE] UPnP AV endpoints (devices and control points) shall be tolerant of DIDL-Lite attributes and elements (whether defined by DIDL-Lite schema or not) that have empty values.

[ATTRIBUTES]

M	C	DMS DMP DMR DMC +DN+ +PU+ +UP+ +PR2+	M-DMS M-DMP M-DMC M-DMU M-DMD	MIU	[46]	QCYRU	

7.4.1.3.5 MM DIDL-Lite Boolean Values

7.4.1.3.5.1

[GUIDELINE] DIDL-Lite Boolean values shall use "0" for false and "1" for true.

[ATTRIBUTES]

M	L	DMS DMR DMC +PU+ +UP+	M-DMS M-DMC M-DMU	n/a	[46]	BWX6P	

NOTE: This simplifies control point implementations and also reduces the size of some UPnP traffic.

7.4.1.3.5.2

[GUIDELINE] UPnP AV endpoints (devices and control points) may parse and interpret DIDL-Lite Boolean values of "yes" and "true" as true ("1"), and "no" and "false" as false ("0"). Such values are interpreted in a case-sensitive manner.

[ATTRIBUTES]

O	A	DMS DMR DMC +PU+ +UP+	M-DMS M-DMC M-DMU	n/a	[46]	YYZDP	

7.4.1.3.6 MM upnp:class Values

7.4.1.3.6.1

[GUIDELINE] UPnP AV MediaServer control point shall minimally treat derived classes in the same way as its ancestor class(es).

[ATTRIBUTES]

M	R	DMP DMC +DN+ +UP+ +PR2+	M-DMP M-DMC M-DMU M-DMD	MIU	[46]	CS3V5	

NOTE: As an example, a UPnP AV MediaServer control point needs to be able to recognize a CDS object marked as an object.item.audioItem.vendorXYZ as an object.item.audioItem even though the UPnP AV MediaServer control point implementation does not understand the meaning behind the vendorXYZ extension. It is not the intent of DLNA to require UPnP AV MediaServer control points to show all CDS objects to a user because some UPnP AV MediaServer control points can be interested in certain types of content classes.

7.4.1.3.6.2

[GUIDELINE] A UPnP AV MediaServer control point shall be tolerant ("parse and interpret" or "parse and ignore") of unknown upnp:class values.

[ATTRIBUTES]

M	R	DMP DMC +DN+ +UP+ +PR2+	M-DMP M-DMC M-DMU M-DMD	MIU	[46]	X7T96	

NOTE: Tolerance of unknown values is required, regardless of whether the control point intends to show the CDS objects to a user.

7.4.1.3.7 MM DIDL-Lite dc:date Format

[GUIDELINE] The syntax for the DIDL-Lite <dc:date> element value shall conform to the following subset profile of [4].

- date-value = date [%x54 time [time-offset]]
- date = 4 DIGIT "-" 2 DIGIT "-" 2 DIGIT ; CCYY-MM-DD
- time = 2 DIGIT ":" 2 DIGIT ":" 2 DIGIT ["." 3 DIGIT"] ; hh:mm:ss(.sss)
- time-offset = %x5a | ("+" | "-") 2 DIGIT ":" 2 DIGIT ; Z or +hh:mm or -hh:mm

Essentially, the following combinations are permitted

- CCYY-MM-DD
- CCYY-MM-DDThh:mm:ss
- CCYY-MM-DDThh:mm:ssZ
- CCYY-MM-DDThh:mm:ss+hh:mm
- CCYY-MM-DDThh:mm:ss-hh:mm
- CCYY-MM-DDThh:mm:ss.sss
- CCYY-MM-DDThh:mm:ss.sssZ
- CCYY-MM-DDThh:mm:ss.sss+hh:mm
- CCYY-MM-DDThh:mm:ss.sss-hh:mm

When the offset of local time to UTC cannot be determined, the <dc:date> string shall have no characters for the <time-offset> part of the date grammar.

[ATTRIBUTES]

M	R	DMS DMR DMC +UP+ +PR2+	M-DMS M-DMC M-DMU		[4] [24]	BYCT4	

7.4.1.3.8 MM DIDL-Lite res@duration Format

[GUIDELINE] The syntax of the res@duration shall be compliant to the following definition.

- duration = hours ":" minutes ":" seconds

- hours = 1*5 DIGIT; 0-99999
- minutes = 2 DIGIT ; 00-59
- seconds = 2 DIGIT ["." 3 DIGIT] ; 00-59 (.000-.999)

[ATTRIBUTES]

M	R	DMC DMS DMR +UP+ +PU+	M-DMS M-DMC M-DMU	MIU	[24]	KFB4S

7.4.1.3.9 MM DIDL-Lite Desc Element Use

7.4.1.3.9.1

[GUIDELINE] The text in the <desc> element , when included in a DIDL-Lite document, shall contain XML-based metadata. This includes the requirement that both the id and nameSpace attributes be included in the <desc> element. The nameSpace attribute shall identify the namespace of the contained XML-based metadata within the <desc> element.

[ATTRIBUTES]

M	R	DMS DMR DMC +PU+ +UP+	M-DMS M-DMC M-DMU	n/a	[46]	EACYX

7.4.1.3.9.2

[GUIDELINE] The XML-based metadata contained within a <desc> element shall have its XML namespace defined within the containing DIDL-Lite document and its value shall be the same as that of the nameSpace attribute.

Shown below are some examples of proper <desc> usage:

EXAMPLE 1:

```
<DIDL-Lite xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/">
   <container id="0000000000000016" searchable="1"
         parentID="000000000000000A" restricted="0" childCount="1">
      <dc:title>some title</dc:title>
      <upnp:class>
         object.item
      </upnp:class>
      <desc id="someid" nameSpace="http://some.example.org/foobar/"
            xmlns="http://some.example.org/foobar/">
         <yada>
            some desc data
         </yada>
      </desc>
   </container>
</DIDL-Lite>
```

EXAMPLE 2:

```
<DIDL-Lite xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/"
      xmlns:ve="http://some.example.org/foobar/">
   <container id="0000000000000016" searchable="1"
         parentID="000000000000000A" restricted="0" childCount="1">
      <dc:title>
         some title
      </dc:title>
      <upnp:class>object.item</upnp:class>
      <desc id="someid" nameSpace="http://some.example.org/foobar/">
```

```
            <ve:yada>
               some desc data
            </ve:yada>
         </desc>
      </container>
   </DIDL-Lite>
```

[Attributes]

M	C	DMS DMR DMC +PU+ +UP+	M-DMS M-DMC M-DMU	n/a	[46] [55]	9O4R8

7.4.1.3.10 MM URI Rules

7.4.1.3.10.1

[Guideline] If a content binary conforms to a DLNA media format profile, then the URI value shall be an absolute URI, with the IP address in the a.b.c.d IPv4 address format (i.e. quad-form network byte order).

[Attributes]

M	L	DMS DMC DMR +PR1+ +PR2+ +PU+ +UP+	M-DMS M-DMC M-DMU	n/a	[49]	Q5KY2

NOTE: This guideline mandates that content URIs will use absolute URIs that use IP addresses.

7.4.1.3.10.2

[Guideline] URIs shall be properly URI-escaped in a UTF-8 encoded form.

[Attributes]

M	L	DMS DMC +PR1+ +PR2+ +PU+ +UP+	M-DMS M-DMC M-DMU	n/a	[19] [46]	G6YNK

NOTE: This guideline requires UPnP AV endpoints to provide URI values that are URI-escaped in a UTF-8 encoded form. This guideline also allows a UPnP AV MediaServer control point to assume that URIs obtained from a UPnP AV MediaServer do not need any escaping.

7.4.1.3.10.3

[Guideline] HTTP URI escaping shall be performed according to the URI specification ([19]) as required in subclause 3.2.1 of the HTTP/1.1 specification ([33]).

[Attributes]

M	L	DMS DMC +PR1+ +PR2+ +PU+ +UP+	M-DMS M-DMC M-DMU	n/a	[19] [33]	UNQQR

7.4.1.3.10.4

[Guideline] URI values that appear in DIDL-Lite documents or fragments shall not exceed 1024 bytes, in the URI-escaped UTF-8 encoded form.

URI values shall not have preceding or trailing white space characters.

[Attributes]

M	L	DMS DMC +PR1+ +PR2+ +PU+ +UP+	M-DMS M-DMC M-DMU	n/a	[19] [46] [55]	NQQRV

NOTE: URI values are theoretically infinite in length. This guideline puts a reasonable limit on the length of advertised content URI values.

7.4.1.3.10.5

[GUIDELINE] If a content binary does not conform to a DLNA media format profile, then the URI value may be a URI, with a domain name.

[ATTRIBUTES]

O	L	DMS DMC +PR1+ +PR2+ +PU+ +UP+	M-DMS M-DMC M-DMU	n/a	[49]	6YNKQ

NOTE: This guideline permits the use of content URIs that use domain names when content is not marked as conformant to a DLNA media format profile. Content sourced from the Internet is considered out of scope for DLNA. However, a ContentDirectory service still has a way of advertising Internet content in a DLNA manner using IPv4 addresses.

7.4.1.3.11 MM DIDL-Lite Recommended Metadata Properties

7.4.1.3.11.1

[GUIDELINE] Content that conforms to the DLNA "Image" media class shall use object.item.imageItem or a derived class for the upnp:class value.

Content that conforms to the DLNA "Audio" media class shall use object.item.audioItem or a derived class for the upnp:class value.

Content that conforms to the DLNA "AV" media class shall use object.item.videoItem or a derived class for the upnp:class value. (A CDS object with the object.item.videoItem or derived class value may have image-based thumbnails as described by 7.1.7 of [56] and 7.2.1 of [56]).

[ATTRIBUTES]

M	A	DMS +UP+	M-DMS M-DMU	n/a	[46] [56]	5KY2U

7.4.1.3.11.2

[GUIDELINE] Metadata properties defined in Appendix B of the ContentDirectory Service specification ([46]) shall use the following namespace prefixes:

Table 18 — Namespace Prefixes

Namespace	Prefix
DIDL-Lite	n/a
Dublin Core	dc:
UPnP	upnp:

[ATTRIBUTES]

M	L	DMS +UP+	M-DMS M-DMU	n/a	[46]	ACYXD

NOTE: n/a means that the prefix is not used.

7.4.1.3.11.3

[GUIDELINE] A UPnP AV MediaServer device should provide non-empty and non-whitespace values for metadata properties as shown in the table below for the purpose of content selection.

Table 19 — Recommended Metadata Properties

upnp:class value	Property Names
object.item.audioItem	dc:creator, upnp:album, upnp:genre, res@duration, res@size
object.item.imageItem	dc:date, res@resolution, res@size
object.item.videoItem	dc:date, upnp:genre, res@duration, res@size
object.container.album.musicAlbum	dc:creator, upnp:genre, @childCount
object.item.videoItem.videoBroadcast object.item.audioItem.audioBroadcast	upnp:genre, upnp:channelName, upnp:channelNr (Applicability of upnp:channelNr depends on region)

This guideline also applies to classes derived from those listed in the table.

As a note, the dc:creator property is understood to contain a value representing the artist name or some equivalent content creator. Whenever possible, the original artist name or content creator should be used for dc:creator. In some cases, such as an audio playlist, a user is the original content creator of the media collection. While it might be appropriate to specify the user's name as the dc:creator for a playlist, it is not recommended to specify the user's name as the dc:creator value for individual audio items in the playlist.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	FB4S5

NOTE: This guideline recommends that some additional metadata properties be used for different media classes. Although not required, providing the user with an information-rich user experience is desirable.

When an Upload Controller or M-DMU creates an object on the MediaServer it will preferably provide these values as part of the DIDL-Lite in the CDS:CreateObject request, see guidelines 7.4.1.7.23.4 and 7.4.1.7.24.2 for this requirement.

7.4.1.3.11.4

[GUIDELINE] If a UPnP AV MediaServer control point lists CDS objects with a particular base-class upnp:class value to users, then it shall also list CDS objects that have a upnp:class value that is derived from the supported base-class.

For example, a MediaServer control point displays content listings for images with a upnp:class value object.item.imageItem needs to also display CDS objects with a upnp:class value of object.item.imageItem.xyz.

[ATTRIBUTES]

M	C	DMP DMC +DN+ +UP+ +PR2+	M-DMP M-DMC M-DMD M-DMU	n/a	[46]	YCT43

NOTE: Control points will not match exclusively on base-class values because guideline 7.4.1.3.11.1 allows MediaServers to use derived-class values. Derived-class values allow a DMS to provide more information about a CDS object. Control points can use this additional information in a variety of ways, but using the upnp:class value to prevent users accessing valid content is not acceptable behavior.

7.4.1.3.12 MM protocolInfo Context

7.4.1.3.12.1

[Guideline] For all guidelines related to protocolInfo values, the following interpretation shall be applied when applying a context for the protocolInfo value. Additions, exceptions, and other clarifications to these baseline interpretation rules shall be indicated in other guidelines on a per-parameter basis.

[Attributes]

M	C	DMP DMS DMR DMC +DN+ +PU+ +UP+ +PR2+	M-DMP M-DMD M-DMS M-DMU M-DMC	MIU	[44] [45] [46]	7T96Y	

7.4.1.3.12.2

[Guideline] The phrase “match by protocolInfo format” shall mean the following.

If given two protocolInfo values, the protocolInfo values “match by protocolInfo format” if they both have the same values for the following.

- first field of protocolInfo, except as noted below.
  - In order to match, equality for the first field is required for all scenarios, except for those involving upload AnyContainer or other optional content management (OCM) operations.
- DLNA.ORG_PN parameter in the fourth field of protocolInfo

[Attributes]

M	C	DMP DMS DMR DMC +DN+ +PU+ +UP+ +PR2+	M-DMP M-DMD M-DMS M-DMU M-DMC	MIU	[44] [45] [46]	S3V59	E

NOTE: This phrase is used throughout the guidelines related to protocolInfo values. The phrase applies to any protocolInfo value, regardless from where the values were obtained. ProtocolInfo values can be found on DMS and DMR devices, typically associated with res@protcolInfo metadata or UPnP AV connection information.

This phrase applies only to protocolInfo values that identify a DLNA media format profile.

7.4.1.3.12.3

[Guideline] If a UPnP AV MediaServer or UPnP AV MediaRenderer exposes a res@protocolInfo with a <res> element that includes a URI value, then the context of the protocolInfo shall be to describe the content and/or transport layer features for the indicated <res> URI value.

[Attributes]

M	C	DMS DMR	M-DMS	n/a	[44] [45] [46]	WX6PS	C

NOTE: DMS devices report res@protocolInfo values when responding to ContentDirectoryService requests, including those related to the Upload System Usage or optional content management operations.

A DMR can create DIDL-Lite metadata to describe what it is currently rendering. See 7.4.1.3.12.12 for more information.

7.4.1.3.12.4

[Guideline] If a UPnP AV MediaServer exposes a res@protocolInfo with a <res> element that omits the URI value but includes a res@importUri property, then the context of the protocolInfo shall be to describe the content that MediaServer expects to receive in a *content transfer process*.

[Attributes]

M	C	DMS	M-DMS	n/a	[45] [46]	CYRUW	C

NOTE: When used with a <res> element that omits a URI value but includes a res@importUri property, the DLNA.ORG_PN parameter identifies the DLNA media format profile that the DMS expects to receive for that particular <res> element. In some cases, DMS implementations will specify various parameters and flags in the 4th field that will become applicable after the content is uploaded and the DMS can serve the content.

7.4.1.3.12.5

[Guideline] If a UPnP AV MediaServer exposes a res@protocolInfo with a <res> element that omits both the URI value and the res@importUri property in a CDS object with the upnp:class property value of object.item.epgItem,object.item.epgItem, object.item.video.videoBroadcast, object.item.audio.audioBroadcast or any of their derived classes, then the context of the protocolInfo shall be to describe the content and/or transport layer features of the content item associated with the CDS object, or of the media resource that the MediaServer expects to expose at a future time.

[Attributes]

M	C	DMS	M-DMS	n/a	[45] [46]	W8JTY	N

NOTE: When used with a <res> element that omits both the URI value and the res@importUri property, the DLNA.ORG_PN parameter identifies the DLNA media format profile of the content resource that the DMS expects to become available later for that particular <res> element. The additional information can allow controllers to make advance arrangements or decisions regarding the content even before it is available. For example, a +SR+ controller can utilize this information to set up a future recording.

7.4.1.3.12.6

[Guideline] If a UPnP AV MediaServer exposes a res@protocolInfo with a <res> element that omits the URI value but includes a res@importUri property, then the protocolInfo 4th field may contain parameters and flags intended for use when the <res> URI value is finally created by the MediaServer.

[Attributes]

O	C	DMS	M-DMS	n/a	[45] [46]	WHVV8	

7.4.1.3.12.7

[Guideline] If a UPnP AV MediaServer lists a protocolInfo in the CMS.SourceProtocolInfo state variable, then the context of the protocolInfo shall be to describe the MediaServer's ability to support the described feature or content for the following:

- serving content to other endpoints on the network

(i.e. The parameters and flags represent a union of the MediaServer's features/capabilities for the specified DLNA media format profile.)

This guideline works in conjunction with 7.4.1.3.13.1.

[Attributes]

M	C	DMS	M-DMS	n/a	[45] [46]	334SX	

NOTE: The phrase “MediaServer's ability to support the described feature or content” means that the UPnP AV MediaServer is capable of serving content binaries that are characterized by the protocolInfo. More specifically, for each protocolInfo listed in SourceProtocolInfo the UPnP AV MediaServer exposes zero or more content binaries that “match by protocolInfo format”. In practice, UPnP AV MediaServers in some cases expose a static list of protocolInfo values to indicate all possible media format profiles that they can serve. In other cases UPnP AV MediaServers will expose a dynamic list of protocolInfo values to indicate the media format profiles that they can currently serve.

For example, if a MediaServer lists "http-get:*:audio/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC;DLNA.ORG_OP=01;DLNA.ORG_FLAGS= AD500000000000000000000000000000" then the MediaServer is capable of serving MPEG_PS_NTSC content with the following characteristics.

- Some of that content might support the Range HTTP header under the "Full Random Access Data Availability" model due to the DLNA.ORG_OP value.
- Some of that content might support the Range HTTP header under the "Limited Random Access Data Availability" model because the lop-bytes flag is true.
- Some of that content might have the sp-flag set to true.
- Some of the content might have beginning point that increases with time (e.g. live content) because the s_0-increasing flag is true.
- Some of the content might have an ending point that increases with time (e.g. infinitely long live content or live content that is currently being saved to disk and will eventually reach a finite length) because the S_N-increasing flag is true.
- Some of the content might support Streaming and Background Transfer modes because the tm-s and tm-b flags are true.

Note that the pn-param (DLNA.ORG_PN) is the only required parameter for DLNA media format profiles. All other parameters and flags are optional for the syntax. (See also 7.4.1.3.14.4.)

7.4.1.3.12.8

[Guideline] If a UPnP AV MediaRenderer lists a protocolInfo in the CMS.SinkProtocolInfo state variable, then the context of the protocolInfo shall be to describe the MediaRenderer's ability to support that the indicated feature or content.

(i.e. The parameters and flags represent a union of the MediaRenderer's features/capabilities for the specified DLNA media format profile).

This guideline works in conjunction with 7.4.1.3.13.1.

[Attributes]

M	C	DMR	n/a	n/a	[44] [45]	34SXQ

NOTE: The phrase “MediaRenderer's ability to support that the indicated feature or content” means that the DMR can use the specified feature when rendering content that is a “match by protocolInfo format”. It is not a guarantee that the DMR will use that feature when rendering such content. Each DMR implementation determines when and how features are used to deliver a playback experience.

For example, if a MediaRenderer lists "http-get:*:audio/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC;DLNA.ORG_OP=01;DLNA.ORG_FLAGS=BD10000000000000 0000000000000000" then the MediaRenderer is capable of rendering MPEG_PS_NTSC content for a variety of scenarios.

- It can render MPEG_PS_NTSC content regardless of whether the sp-flag, s_0-increasing, or S_N-increasing flags are set to true or false for that content.
- If the Content Source makes the Range HTTP header available, the Content Receiver is able to use the Range HTTP header for both "Full Random Access Data Availability" and "Limited Random Access Data Availability" models because the op-param and the lop-bytes flag.
- If given a PlayContainer URI, the DMR will also play MPEG_PS_NTSC content found during the traversal of the DMS because the playcontainer-param is set to true.
- The DMR uses Streaming transfers for the MPEG2_PS_NTSC content, as required for immediate rendering

Note that the pn-param (DLNA.ORG_PN) is the only required parameter for DLNA media format profiles. All other parameters and flags are optional for the syntax.

7.4.1.3.12.9

[Guideline] If a UPnP AV MediaServer or UPnP AV MediaRenderer uses a protocolInfo value in the *ProtocolInfo* output argument of a CMS:GetCurrentConnectionInfo response, then the context of the protocolInfo shall be to describe the content and the transport layer capabilities of the UPnP AV connection.

[ATTRIBUTES]

M	C	DMS DMR	M-DMS	n/a	[44] [45] [46]	HVV8S

NOTE: In this context, the protocolInfo value describes the content and/or transport layer capabilities of the UPnP AV connection from the server-side.

For example, if a UPnP AV connection indicates "http-get:*:audio/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC;DLNA.ORG_OP=01;DLNA.ORG_FLAGS=0510000000000000 0000000000000000" then the transport layer associated with the UPnP AV connection is configured to work in the following ways.

- The MPEG_PS_NTSC content has the sp-flag set to false and the tm-s flag is set to true to indicate that transport layer is using a Streaming transfer where the Content Source is not the Clock Source. This means the Content Source will ensure that it transmits at a throughput necessary for rendering but will also preserve the entire content bitstream even for lower transmission throughputs when network conditions are not able to support the bandwidth needed for the Streaming transfer.
- The Range HTTP header is available only under the "Full Random Access Data Availability" model.
- The content is growing with a fixed beginning because the s_0-increasing flag is false and the s_N-increasing flag is true.

The phrase “describe the content and the transport layer capabilities of the UPnP AV connection” does not imply that actual data is actually being transported on the UPnP AV connection at the current moment because the Content Receiver endpoint might be in a playback state (such as stop or pause) that does not involve the ongoing transmission of content data.

7.4.1.3.12.10

[GUIDELINE] If a UPnP AV MediaServer control point creates a res@protocolInfo value for use with a CDS:CreateObject request that qualifies as one of the listed operations, then the context of the protocolInfo shall be to describe the content that is going to be uploaded to the MediaServer through a content transfer process.

- upload AnyContainer
- OCM: upload content

Furthermore, the default behavior of the control point shall be to exclude 4th field parameters (except the DLNA.ORG_PN parameter, described in 7.4.1.3.17) or specify the parameter with a false value (such as in the case of the lop-npt flag, described in 7.4.1.3.28).

[ATTRIBUTES]

M	C	+UP+	M-DMU	MIU	[46]	YRUW7

NOTE: Unless specified otherwise, 4th field parameters do not apply to what the Upload Controller can do. Rather, the DMS will provide the appropriate 4th field parameters during the course of an Upload System Usage or an optional content management operation. See 7.4.1.3.12.4 and 7.4.1.3.12.6 for more information.

7.4.1.3.12.11

[GUIDELINE] If a UPnP AV MediaRenderer control point creates a protocolInfo or res@protocolInfo value, then the context of the protocolInfo shall be to describe the Content Source's capabilities for the associated UPnP AV connection or <res>.

Note: “MediaRenderer control point creates a protocolInfo or res@protocolInfo” specifically refers to the following scenarios. This phrase is used in other guidelines related to protocolInfo.

- res@protocolInfo values that appear in CDS or DIDL-Lite metadata used as input arguments for AVT:SetAVTransportURI, or other UPnP AV actions.

[ATTRIBUTES]

M	C	DMC +PU+	M-DMC	MIU	[44] [45] [46]	X6PSO

NOTE: In the case of AVT:SetAVTransportURI, the res@protocolInfo can appear in the DIDL-Lite metadata of the *CurrentURIMetaData* input argument.

7.4.1.3.12.12

[GUIDELINE] A UPnP AV MediaRenderer may use res@protocolInfo values in the following ways.

- res@protocolInfo values may appear in any AVTransport virtual instance state variable, including but not limited to: AVT.NextAVTransportURIMetaData, AVT.CurrentTrackMetaData, and AVT.AVTransportURIMetaData.
- res@protocolInfo values may appear in output arguments of any AVTransport actions, including but not limited to:
  - *CurrentURIMetaData* of AVT:GetMediaInfo
  - *NextURIMetaData* of AVT:GetMediaInfo, or
  - *TrackMetadata* of AVT:GetPositionInfo.

[ATTRIBUTES]

O	C	DMR	n/a	n/a	[44]	3V59O

NOTE: This guideline clarifies where a DMR can use res@protocolInfo values.

7.4.1.3.13 MM Consistency of protocolInfo State Variables and Output Arguments

7.4.1.3.13.1

[GUIDELINE] If the listed values for the CMS.SinkProtocolInfo or CMS.SourceProtocolInfo state variables change, then the ConnectionManager service shall report the change through the GENA event mechanism, as defined by [47].

CMS:GetProtocolInfo shall return the same protocolInfo list for *SinkProtocolInfo* output argument as the current list indicated by the CMS.SinkProtocolInfo state variable.

CMS:GetProtocolInfo shall return the same protocolInfo list for *SourceProtocolInfo* output argument as the current list indicated by the CMS.SourceProtocolInfo state variable.

[ATTRIBUTES]

M	R	DMS DMR	M-DMS	n/a	[47] [45]	T96YP

NOTE: Many 4th_field DLNA guidelines refer to CMS:GetProtocolInfo arguments, without referring specifically to the associated ConnectionManager service state variables. This guideline repeats the ConnectionManager service's requirement that CMS:GetProtocolInfo accurately represent the data in the associated state variables, which in turn allows the DLNA syntax rules to apply to CMS state variables.

7.4.1.3.14 MM CMS:GetProtocolInfo Rules

7.4.1.3.14.1

[GUIDELINE] The ConnectionManager service of a UPnP AV MediaServer shall list the union set of protocolInfo values supported by the device for protocolInfo values that share the same values in the first three fields and the same pn-param (DLNA.ORG_PN) value in the fourth field, but have different values in the additional parameters in the fourth field. This means that the ConnectionManager service shall list only one protocolInfo value to represent all such profiles.

The first three fields of the listed protocolInfo value shall be identical to the first three fields of the individual protocolInfo values. The pn-param (DLNA.ORG_PN) value in the fourth field shall be identical to the pn-param (DLNA.ORG_PN) value in the fourth fields of these protocolInfo values.

If additional parameters are included in the fourth field of the listed protocolInfo value, they shall appear in the order defined in 7.4.1.3.16, and their values shall be obtained as defined in 7.4.1.3.14.6.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[45]	CT43W	

NOTE: This guideline makes it easier for control points to find servers that provide content for a given profile.

Note that use of the 'http-get:*:*:*' protocolInfo does not sufficiently communicate the supported media format profiles. This guideline also requires implementations to explicitly list the individual protocolInfo values for supported DLNA media format profiles.

7.4.1.3.14.2

[GUIDELINE] The sets of protocolInfo values returned in CMS:GetProtocolInfo shall list the protocolInfo values that use the http-get media transport and DLNA media format profiles first.

[ATTRIBUTES]

M	L	DMS DMR	M-DMS	n/a	[45]	B4S5U	

NOTE: Although the number of returned protocolInfo values might be large, control points can use this guideline to capture the subset of relevant DLNA protocolInfo values.

7.4.1.3.14.3

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						CYXD3	

7.4.1.3.14.4

[GUIDELINE] The fourth field of the protocolInfo values (obtained from the ConnectionManager service) that have the pn-param parameter in the fourth field shall follow one of the formatting conventions.

- In addition to pn-param, provide one or more of the additional parameters as defined in guideline 7.4.1.3.16.1 (HTTP), or guideline 7.4.1.3.16.2 (RTP/RTSP)
- Provide only the pn-param parameter, as defined in guideline 7.4.1.3.17

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[45]	Y8Y9W	

NOTE: This guideline allows UPnP AV MediaServer implementations to employ a verbose listing that includes trick/seek mode capabilities or simply provide a listing based on media format profiles.

This guideline also allows UPnP AV MediaRenderer implementations to employ only primary protocolInfo sets (non-verbose listing), or in addition, to extend the list of declared 4th field parameters (verbose listing).

7.4.1.3.14.5

[GUIDELINE] If a CSV (Comma Separated Value) list contained in a CMS:GetProtocolInfo response has one or more embedded comma(s) in the individual substring entries of the CSV list, then those embedded commas shall be escaped as "\," in [46] subclause 2.3.1.

[ATTRIBUTES]

M	A	DMS DMR	M-DMS		[45] [46]	WYAJV	

NOTE: There is no embedded comma escaping rules in [45]. This guideline applies embedded comma escaping rules in CSV lists defined in [46] to CMS:GetProtocolInfo response values.

7.4.1.3.14.6

[GUIDELINE] If a UPnP AV MediaServer lists additional parameters (besides the pn-param) in the 4th field of protocolInfo when combining multiple protocolInfo values as defined in 7.4.1.3.14.1, then they shall be obtained as follows:

- op-param: If none of the individual protocolInfo values contain an op-param value, the combined protocolInfo value shall omit the op-param value. If the a-val of the op-param in any of the individual protocolInfo values is "1", the a-val of the op-param in the combined protocolInfo value shall be "1". Otherwise, the a-val of the op-param in the combined protocolInfo value shall be "0". Similarly, if the b-val of the op-param in any of the individual protocolInfo values is "1", the b-val of the op-param in the combined protocolInfo value shall be "1". Otherwise, the b-val of the op-param in the combined protocolInfo value shall be "0".
- ps-param: If none of the individual protocolInfo values contain a ps-param value, the combined protocolInfo value shall omit the ps-param value. Otherwise, the ps-param value of the combined protofocolInfo value shall be a comma-separated list of all the distinct ps-param values of the individual protocolInfo values.
- ci-param: If none of the individual protocolInfo values contain a ci-param value, the combined protocolInfo value shall omit the ci-param value. If the ci-param value in any of the individual protocolInfo values is "1", the ci-param value in the combined protocolInfo value shall be "1". Otherwise, the ci-param value in the combined protocolInfo value shall be "0".
- flags-param: If none of the individual protocolInfo values contain a flags-param value, the combined protocolInfo value shall omit the flags-param value. For each bit in the flags-param value, if the corresponding bit in any of the individual protocolInfo values is "1", that bit in the combined protocolInfo value shall be "1". Otherwise, that bit shall be "0".
- maxsp-param: If none of the individual protocolInfo values contain a maxsp-param value, the combined protocolInfo value shall omit the maxsp-param value. Otherwise, the maxsp-param value of the combined protocolInfo value shall be the maximum of all the maxsp-param values of the individual protocolInfo values.
- other-param: If none of the individual protocolInfo values contain an other-param value, the combined protocolInfo value shall omit the other-param value. Otherwise, the combined protocolInfo value shall contain the concatenated list of all the distinct other-param values in individual protocolInfo values.

Note that the combined protocolInfo value might not be a valid res@protocolInfo value for content items in a content directory. For example, it might contain contradicting flags which are prohibited by the DLNA guidelines.

For example, the union set of the following protocolInfo values

- http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC;DLNA.ORG_OP=01; DLNA.ORG_PS=2
- http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC;DLNA.ORG_OP=10
- http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC;DLNA.ORG_PS=4

is either one of the following (depending on whether additional fourth-field parameters are included)

- http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC

- http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC;DLNA.ORG_OP=11;DLNA.ORG_PS=2,4

[Attributes]

M	R	DMS	M-DMS	n/a	[45]	928RT

7.4.1.3.14.7

[Guideline] The ConnectionManager service of a UPnP AV MediaRenderer shall list the collection of distinct Primary protocolInfo Sets that it is capable of rendering.

If one or more of the declared Primary protocolInfo Sets include additional 4th field parameters, they shall appear in the order defined in 7.4.1.3.16, and their values shall be as defined in 7.4.1.3.14.8

[Attributes]

M	R	DMR	n/a	n/a	[45]	ETV64

7.4.1.3.14.8

[Guideline] If a UPnP AV MediaRenderer lists additional parameters (besides the pn-param) in the 4th field of protocolInfo when declaring the Primary protocolInfo Sets as defined in 7.4.1.3.14.7, then they shall comply with the following rules:

- op-param: This parameter shall not be included.
- ps-param: This parameter shall not be included.
- ci-param: This parameter shall not be included.
- flags-param: This parameter shall be included when the UPnP AV MediaRenderer is capable of rendering this type of content in a PlayContainer URI operation.
- maxsp-param: This parameter shall not be included.
- other-param: The inclusion or omission of this vendor-defined parameter is vendor dependent.

[Attributes]

M	R	DMR	n/a	n/a	[45]	TV648

7.4.1.3.15 MM DIDL-Lite protocolInfo values

7.4.1.3.15.1

[Guideline] If the <res> value contains an HTTP URL, then the first field of the res@protocolInfo value shall be as shown below.

- http-get

[Attributes]

M	R	DMS DMR +PU+	M-DMS	n/a	[46]	KY2U3

NOTE: Guidelines in 7.4.1.3.15 provide requirements on the protocolInfo values in DIDL-Lite documents.

For requirements around protocolInfo values used in the upload process, see guideline 7.4.1.7.19.1.

Reiterates the protocol string value for DLNA content transported across HTTP. Note that the converse is not always true. Some scenarios involving upload operations can result with a <res> element that has no value (i.e. empty URI) but the res@protocolInfo value has "http-get" in the first field.

7.4.1.3.15.2

[GUIDELINE] If the <res> value contains an RTP URL, then the first field of the res@protocolInfo value shall be as shown below.

- rtsp-rtp-udp

[ATTRIBUTES]

M	R	DMS DMR +PU+	M-DMS	n/a	[46]	YNKQ9	

7.4.1.3.15.3

[GUIDELINE] The third field of a protocolInfo value shall use the MIME types specified in caluse 5 of [56] for DLNA normative media format profiles when the protocol is "http-get" or "rtsp-rtp-udp".

[ATTRIBUTES]

M	F	DMS DMR DMC +PU+	M-DMS M-DMC	n/a	[46] [56]	QQRVR	

NOTE: This guideline is consistent with the UPnP AV specifications for the case of "http-get", but this guideline also conflicts in the case of "rtp-rtsp-udp".

The UPnP AV specification indicates that the 4th field for RTP URIs is the payload type, but such a model fails to take into account that streaming of AV content with RTP generally requires multiple RTP payload types. For this reason, the DLNA guidelines require that the 3rd field is populated with the DLNA-specified mime-type of the media format profile indicated in the pn-param.

7.4.1.3.15.4

[GUIDELINE] UPnP AV endpoints (devices and control points) shall use the DLNA.ORG_PN parameter (7.4.1.3.17) in the 4th field of protocolInfo for the following operations:

- To identify content conforming to a DLNA media format profile, or
- To specify that a UPnP device is capable of receiving, distributing, or rendering content conforming to a DLNA media format profile.

[ATTRIBUTES]

M	L	DMS DMP DMR DMC +DN+ +PU+ +UP+ +PR2+	M-DMP M-DMS M-DMD M-DMU M-DMC	MIU	[46]	QRVR5	

NOTE: MIME-types in the 3rd field of res@protocolInfo alone are not sufficient to identify DLNA content.

MIME-types are not to be used to identify DLNA media format profiles because MIME-types will be updated by IANA.

7.4.1.3.15.5

[GUIDELINE] If a protocolInfo value has the first value of "http-get" and the 4th field includes the DLNA.ORG_PN parameter (7.4.1.3.17), then the binary data conformant to a DLNA media format profile identified by the DLNA.ORG_PN parameter shall be transmitted via the HTTP Media Transport, without transport layer encryption.

[ATTRIBUTES]

M	L	DMS DMP DMR DMC +DN+ +PU+ +UP+ +PR2+	M-DMP M-DMS M-DMD M-DMU M-DMC	MIU	[45]	NKQ9V	

NOTE: This means that the binary data via the HTTP Media Transport is not encrypted unless the guidelines of the DLNA media format profile explicitly states it.

EXAMPLE 1: The following are example scenarios of creating 4th field values, and the DLNA device classes capabilities that apply to them.

[C1] DMS, M-DMS, MIU: These endpoints populate res@protocolInfo in metadata responses and protocolInfo used in CMS:GetProtocolInfo.

[C2] DMS, M-DMS, +PU+ +PR1+, MIU: These endpoints populate values for contentFeatures.dlna.org at the transport layer, either by creating a brand new 4th field value or modifying from existing 4th field value.

[C3] DMC, M-DMC, +PU+, MIU: The control points of these endpoints can create a 4th field value for metadata given to a DMR, either by creating a brand new 4th field value or modifying an existing 4th field value.

[C4] DMR, MIU: When reporting metadata of content that is currently playing, these endpoints can create a new 4th field value by modifying an existing 4th field value acquired from a DMS, DMC, or +PU+.

[C5] M-DMU, +UP+, MIU: These endpoints create res@protocolInfo values in CDS:CreateObject requests and also create the contentFeatures.dlna.org value during the content transfer process.

EXAMPLE 2: The following are example scenarios of parsing 4th field values, and the DLNA device classes capabilities that apply to them.

[P1] DMP, M-DMP, M-DMD, DMR, DMC, M-DMC, M-DMU, +UP+, +PR2+, +DN+, MIU: Control points of these endpoints parse 4th field values in metadata responses obtained from a DMS or M-DMS.

[P2] DMP, M-DMP, M-DMD, DMR, DMPr, +DN+, MIU: These endpoints parse 4th field values from contentFeatures.dlna.org, at the transport layer.

[P3] DMR, MIU: These endpoints can receive 4th field values in DIDL-Lite metadata, provided by a control point.

[P4] DMC, M-DMC, +PU+, MIU: The control points of these endpoints can receive 4th field values in DIDL-Lite metadata that is exposed by a DMR.

[P5] DMS, M-DMS, MIU: These endpoints parse res@protocolInfo values in CDS:CreateObject requests and also parse the contentFeatures.dlna.org value during the content transfer process.

7.4.1.3.16 MM protocolInfo values: 4th Field

7.4.1.3.16.1

[GUIDELINE] If a protocolInfo value has "http-get" as the first field value and the 4th field includes the pn-param token, then the following syntax shall be used for the fourth field.

- 4th_field = pn-param [op-param] [ps-param] [ci-param] [flags-param] [*(other-param)]

Note: In all guidelines that relate to the syntax of the 4th field, the relative order of pn-param, op-param, ps-param, ci-param, flags-param, and *(other-param) used in 4th_field is mandatory. For example, pn-param cannot appear after op-param, ps-param, ci-param, flags-param, or *(other-params).

The syntax and definition of pn-param, op-param, ps-param, ci-param, flags-param, and *(other-param) are defined in the guidelines below.

[ATTRIBUTES]

M	A	DMP DMS DMC DMR DMPr +DN+ +UP+ +PU+ +PR1+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	Y2U3H

NOTE: This guideline defines the syntax of the fourth field for a res@protocolInfo value that indicates content that is transported across HTTP.

Note that this syntax prohibits the use of the "*" value for content that conforms to a DLNA media format profile. Content that does not conform to a DLNA media format profile can use the "*" value in the 4th field.

The following examples apply to this guideline: [C1], [C2], [C3], [C4], [P1], [P2], [P3], and [P4].

Please note that examples [C5] and [P5] do not apply to this guideline because 7.4.1.7.19 covers those cases.

7.4.1.3.16.2

[Guideline] If a protocolInfo value has "rtsp-rtp-udp" as the first field value and the 4th field includes the pn-param token, then the following syntax shall be used for the fourth field.

- 4th_field-rtp = pn-param [op-param] [ps-param] [ci-param] [flags-param] [maxsp-param] [*(other-param)]

[Attributes]

M	A	DMS DMP DMR DMC +DN+ +PU+ +UP+ +PR2+	MHD	MIU	[44] [45] [46]	YXD3U

NOTE: The 4th field syntax for rtsp-rtp-udp URIs is the same as for http-get URIs.

The following examples apply to this guideline: [C1], [C2], [C3], [C4], [P1], [P2], [P3], and [P4].

Note that DMPr and +PR1+ are not listed for this guideline because RTP Media Transport is never used to transport images. However M-DMD, +DN+, and +PR2+ are listed because the associated control points are required to tolerate <res> elements that use the RTP Media Transport when browsing a MediaServer.

7.4.1.3.16.3

[Guideline] If a protocolInfo value has the first field value that is neither equal to "http-get" nor "rtsp-rtp-udp", then the fourth field may have a syntax that differs from 7.4.1.3.16.1 and 7.4.1.3.16.2.

[Attributes]

O	A	DMP DMS DMC DMR DMPr +DN+ +UP+ +PU+ +PR1+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	4S5UX

NOTE: This guideline permits a different syntax for the fourth field of a protocolInfo value when the content is not transported over a DLNA-specified transport.

The following examples apply to this guideline: [C1], [C2], [C3], [C4], [P1], [P2], [P3], and [P4].

Note that the CDS:CreateObject syntax has restrictions on the 4th field syntax, as described in 7.4.1.7.19.

Examples of protocolInfo values that include the 4th field are shown below;

- http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC
- http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_PAL;DLNA.ORG_OP=10
- http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC;DLNA.ORG_OP=10; DLNA.ORG_PS=-1,30,-30
- http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_PAL;DLNA.ORG_PS=-1,30,-30
- http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_PAL; DLNA.ORG_PS=-1,30,-30;VENDORXYZ.COM_FOO=bar
- http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_PAL;DLNA.ORG_CI=1; VENDORXYZ.COM_FOO=bar
- rtsp-rtp-udp:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC;DLNA.ORG_OP=10; DLNA.ORG_PS=-1,2/3,4;DLNA.ORG_FLAGS=03100000000000000000000000000000; DLNA.ORG_MAXSP=9.75

7.4.1.3.17 MM pn-param (DLNA.ORG_PN Parameter)

[Guideline] The syntax definition of pn-param shall be as follows:

- pn-param = "DLNA.ORG_PN=" pn-value
- pn-value = *<"a"-"z", "A"-"Z", "0"-"9", "_">

The pn-value shall identify the DLNA media format profile ID that is applicable for the context of the protocolInfo. (See 7.4.1.3.12 for determining an appropriate context for the protocolInfo.)

The pn-param is reserved for use with contexts where content conforms to a DLNA media format profile. Use of pn-param for content not conformant with a DLNA media format profile is expressly prohibited.

[ATTRIBUTES]

M	A	DMP DMS DMC DMR DMPr +DN+ +UP+ +PU+ +PR1+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	T43WJ

NOTE: This guideline defines the syntax of the DLNA.ORG_PN parameter. This parameter is used to identify DLNA content. This parameter cannot be used for non-DLNA content.

The following examples apply to this guideline: [C1], [C2], [C3], [C4], [C5], [P1], [P2], [P3], [P4], and [P5].

7.4.1.3.18 MM op-param (Operations Parameter - Common Guidelines)

7.4.1.3.18.1

[GUIDELINE] The syntax definition of op-param shall be as follows:

- op-param = [op-param-delim] "DLNA.ORG_OP=" op-value
- op-param-delim = ";"
- op-value = a-val b-val
- a-val = Boolean
- b-val = Boolean
- Boolean = "1" | "0"

The op-value is a string composed of two characters: a-val and b-val. The meaning of these values is described in these guidelines, depending on whether the context is for the HTTP Media Transport or RTP Media Transport.

- 7.4.1.3.19 MM op-param (Operations Parameter for HTTP)
- 7.4.1.3.20 MM op-param (Operations Parameter for RTP)

If the first field of protocolInfo is neither "http-get" nor "rtsp-rtp-udp" then the fourth field shall omit the op-param.

[ATTRIBUTES]

M	A	DMP DMS DMC DMR DMPr +DN+ +UP+ +PU+ +PR1+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	96YPS

NOTE: This guideline defines the DLNA.ORG_OP parameter.

The following examples apply to this guideline: [C1], [C2], [C3], [C4], [P1], [P2], [P3], and [P4].

7.4.1.3.18.2

[GUIDELINE] If the op-param is present and if either a-val or b-val is "1", then the "Full Random Access Data Availability" model shall be the data access model that applies in the context of the protocolInfo value.

Specifically, this means that the transport operation (that is indicated by the a-val or b-val) shall be supported for the entire content binary, as defined in 7.5.4.2.15.1.

If the flags-param token is included in the 4th field, then the s_0-increasing, lop-npt, and lop-bytes bits of the primary-flags token shall be set to false.

[ATTRIBUTES]

M	C	DMS +PU+ +PR1+	M-DMS	MIU	[44] [45] [46]	V59OT	

NOTE: The DLNA.ORG_OP parameter indicates support for transport layer headers responsible that facilitate random access operations on content binaries, under the "Full Random Access Data Availability" model.

For more information on the "Full Random Access Data Availability" model, see the following guidelines.

- 7.5.4.2.14 MT Normative Random Access Data Availability Models
- 7.5.4.2.15 MT "Full Random Access Data Availability" Model

The "Full Random Access Data Availability" model is mutually exclusive with the "Limited Random Access Data Availability" model, which is why lop-npt/lop-bytes cannot be used with the op-param. For more information, see the following guidelines.

- 7.4.1.3.28 MM lop-npt, lop-bytes and lop-cleartextbytes (Limited Operations Flags): Common
- 7.5.4.2.16 MT "Limited Random Access Data Availability" Model

The following examples apply to this guideline: [C1] and [C2].

7.4.1.3.18.3

[GUIDELINE] In conjunction with the rules defined in 7.4.1.3.18.1 and 7.4.1.3.18.2, the fourth field of a protocolInfo may use the op-param for non-DLNA media format profiles.

[ATTRIBUTES]

O	A	DMS +PU+ +PR1+	M-DMS	MIU	[44] [45] [46]	6PSOW	

NOTE: This guideline permits the use of DLNA.ORG_OP for both DLNA and non-DLNA content, provided the DLNA-defined syntax and semantics are used.

The following examples apply to this guideline: [C1] and [C2].

7.4.1.3.18.4

[GUIDELINE] If the b-val token is true and s_N-increasing flag is false, then the Content Source shall provide content length information in the res@size value of the CDS object.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	MIU	[46]	RUW7R	C

NOTE: This guideline helps resolve dependency issues that Content Receivers have on knowing the content length, and where Content-Length is not provided in a chunked response. Content Sources are also encouraged to provide a res@size value even in scenarios where byte Range is not supported.

The following examples apply to this guideline: [C1] and [C2]. Note that this guideline does not apply to +PR1+ because it does not advertise content with res@size.

7.4.1.3.18.5

[GUIDELINE] If the a-val token is true and s_N-increasing flag is false, then the Content Source shall provide duration information in the res@duration value of the CDS object.

[Attributes]

M	L	DMS +PU+	M-DMS	MIU	[46]	VV8SU	C

NOTE: This guideline resolves dependency issues that Content Receivers have on knowing the content duration. If Content Sources support TimeSeekRange.dlna.org, duration information is important, and Content Sources are urged to provide it.

The following examples apply to this guideline: [C1] and [C2]. Note that this guideline does not apply to +PR1+ because it does not advertise content with res@size.

7.4.1.3.18.6

[Guideline] If the Pause media operation is supported, Rendering Endpoints shall support the Pause Release media operation even in the absence of Content-Length, res@size, or res@duration information.

[Attributes]

M	L	DMR DMP	M-DMP	MIU	[46]	4SXQQ	

NOTE: Rendering Endpoints can issue a Range request using the "beginrange-" notation so as to avoid specifying an invalid range. Rendering Endpoints can issue a TimeSeekRange.dlna.org request using the "begintime-" notation so as to avoid specifying an invalid range.

The variants of examples [P1] and [P2] that involve rendering of audio or AV content apply to this guideline.

7.4.1.3.19 MM op-param (Operations Parameter for HTTP)

7.4.1.3.19.1

[Guideline] If the 4th field is associated with the "http-get" transport protocol, then the a-val and b-val tokens (of the op-param token) mean the following.

- a-val: indicates support of the TimeSeekRange.dlna.org HTTP header (see 7.5.4.3.2.24) for the context of the protocolInfo under the "Full Random Access Data Availability" model
- b-val: indicates support of the Range HTTP header (see 7.5.4.3.2.22) for the context of the protocolInfo under the "Full Random Access Data Availability" model

[Attributes]

M	A	DMS +PU+ +PR1+	M-DMS	MIU	[33], [45] [46]	SXQQ6	

•

NOTE: This guideline defines the op-param token's a-val and b-val tokens when HTTP is the transport protocol.

When used with a context involving HTTP, the DLNA.ORG_OP parameter identifies if the server supports the TimeSeekRange.dlna.org or Range HTTP headers for the associated content binary under the "Full Random Access Data Availability" model.

For more information on the "Full Random Access Data Availability" model for HTTP and these HTTP headers, see the following guidelines.

- 7.5.4.2.14 MT Normative Random Access Data Availability Models
- 7.5.4.2.15 MT "Full Random Access Data Availability" Model
- 7.5.4.3.2.18 MT HTTP Common Random Access Data Availability Requirements
- 7.5.4.3.2.19 MT HTTP Data Range of "Full Random Access Data Availability"

The following examples apply to this guideline: [C1] and [C2]. Note that the TimeSeekRange.dlna.org header never applies to scenarios involving XHTML print documents and images.

7.4.1.3.19.2

[Guideline] If the associated HTTP Server Endpoint returns HTTP error code 406 (Not Acceptable) because an HTTP specifies one of the above HTTP headers, then the op-param shall indicate that the HTTP header is not supported by using a value of "0" for the appropriate a-val or b-val.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+	M-DMS	MIU	[33], [45] [46]	V8SUN	

NOTE: HTTP Server Endpoints use the 406 (Not Acceptable) status code to indicate that an HTTP request can never be satisfied with the specified HTTP headers.

The following examples apply to this guideline: [C1] and [C2].

7.4.1.3.19.3

[GUIDELINE] If the associated HTTP Server Endpoint always returns 406 (Not Acceptable) in response to requests that use either HTTP header for the context of the protocolInfo, then the 4th field shall omit the op-param.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+	M-DMS	MIU	[33], [45] [46]	UW7RA	

NOTE: Including an op-param with a value of "00" is prohibited. In cases where neither the TimeSeekRange.dlna.org nor the Range header are supported, the op-param is omitted.

The following examples apply to this guideline: [C1] and [C2].

7.4.1.3.19.4

[GUIDELINE] If the associated HTTP Server Endpoint is capable of responding with a Target Response that appropriately corresponds to the data range indicated in the HTTP request's TimeSeekRange.dlna.org header value, then the a-val token shall be true.

(i.e. The associated HTTP Server Endpoint does not return error code 406 (Not Acceptable), unless other HTTP headers are the cause of the error.)

Lastly, the Content Source shall be able to support TimeSeekRange.dlna.org on the entire content binary, as required by 7.4.1.3.18.2.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+	M-DMS	MIU	[33], [45] [46]	PSOWQ	

NOTE: An HTTP Server that can respond with content data occupying a particular npt time range needs to advertise this capability in the 4th field.

The following examples apply to this guideline: [C1] and [C2].

Note that "appropriately corresponds" is clarified by 7.5.4.3.2.24.6 (MT HTTP Time-Based Seek (Server)), which permits eturning data from a decoder-friendly point.

7.4.1.3.19.5

[GUIDELINE] If the associated HTTP Server Endpoint is capable of responding with a Target Response that corresponds exactly to the data range indicated in the HTTP request's Range header value, then the b-val token shall be true.

(i.e. The associated HTTP Server Endpoint does not return error code 406 (Not Acceptable), unless other HTTP headers are the cause of the error.)

Lastly, the Content Source shall support Range on the entire content binary, as required by 7.4.1.3.18.2.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+	M-DMS	MIU	[33], [45] [46]	59OTH	

NOTE: An HTTP Server that can respond with content data occupying a particular byte range needs to advertise this capability in the 4th field.

The following examples apply to this guideline: [C1] and [C2].

7.4.1.3.19.6

[GUIDELINE] If the associated HTTP Server Endpoint supports the realTimeInfo.dlna.org HTTP header with a finite max-lag-time value (as described in 7.5.4.3.3.19.2), then

- the op-param shall be omitted
- sp-flag = true
- lop-bytes = false
- lop-npt = false

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	[33], [45] [46]	6YPST	

NOTE: The following examples apply to this guideline: [C1] and [C2]. Note that this guideline never applies to printing related scenarios, which is why +PR1+ is excluded from this guideline.

7.4.1.3.20 MM op-param (Operations Parameter for RTP)

7.4.1.3.20.1

[GUIDELINE] If the 4th field is associated with the "rtsp-rtp-udp" transport protocol, then the a-val and b-val tokens (of the op-param token) mean the following.

- a-val: indicates support of the Range header (see 7.5.4.4.6.2.40 and 7.5.4.4.6.2.41) for the context of the protocolInfo under the "Full Random Access Data Availability" model
- b-val: In scenarios involving the RTP Media Transport, the b-val shall have a value of "0".

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	n/a	43WJU	

NOTE: For more information about the RTP Media Transport and the "Limited Random Access Data Availability" model, see the following guidelines.

- 7.5.4.2.14 MT Normative Random Access Data Availability Models
- 7.5.4.2.15 MT "Full Random Access Data Availability" Model
- 7.5.4.4.6.2.40 MT RTP Receiving Endpoint Range header
- 7.5.4.4.6.2.41 MT RTP Serving Endpoint Range header

Excluding variants related to printing, the following examples apply to this guideline: [C1] and [C2].

7.4.1.3.20.2

[GUIDELINE] If the Content Source assigns "0" to both a-val and b-val of the op-param, then the op-param shall be omitted from the 4th field.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+	M-DMS	MIU	[33], [45] [46]	S5UX6	E

NOTE: The following examples apply to this guideline: [C1] and [C2].

7.4.1.3.21 MM ps-param (Server-Side PlaySpeeds Parameter)

7.4.1.3.21.1

[GUIDELINE] The definition of ps-param shall be as follows:

- ps-param = [ps-param-delim] "DLNA.ORG_PS=" ps-value
- ps-param-delim = ";"
- ps-value = [server-speed *("," server-speed)]
- server-speed = <conforms to the TransportPlaySpeed string, as specified in the AVTransport specification>

The ps-value shall be a comma-delimited list of play speed values. The ps-value shall exclude the play speed of "1" from its list. If the media transport component (either for a server, client) does not support additional server-side play speeds beyond "1" for the context of the protocolInfo, then the fourth field shall omit the ps-param (i.e. "DLNA.ORG_PS=1" is prohibited).

The format of each play speed value shall conform to the TransportPlaySpeed string, as specified in [44], 2.2.8.

If used in conjunction with a protocolInfo indicating "http-get" in the first field, then the use of the PlaySpeed.dlna.org HTTP header applies to the context of the protocolInfo. See 7.5.4.3.3.16 for more information.

If used in conjunction with a protocolInfo indicating "rtsp-rtp-udp" in the first field, then the use of the SCALE RTSP header applies to the context of the protocolInfo. See 7.5.4.4.6.2.42 for more information.

If using the 4th_field syntax defined in 7.4.1.3.16.1 or 7.4.1.3.16.2 and if ps-param follows another parameter, then the ps-param shall include the ps-param-delim.

[ATTRIBUTES]

M	A	DMP DMS DMC DMR +DN+ +UP+ +PU+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	XD3U5	E

NOTE: This guideline defines the DLNA.ORG_PS parameter. The parameter indicates the transport layer's supported play speeds.

Excluding variants related to printing, the following examples apply to this guideline: [C1], [C2], [C3], [C4], [P1], [P2], [P3], and [P4]. The ps-param is not used for image content nor XHTML print documents, but control points of printing-related endpoints still need to tolerate the presence of the ps-param.

A ps-value can have "," characters as delimiters for supporting server-side play speeds and the DLNA guidelines defines a embedded comma escaping rule for a value of a CMS:GetProtocolInfo response. Refer to 7.4.1.3.14.5 for more information.

Each of the play speed values in ps-param is listed as an integer or as a ratio of two integer values. It is highly recommended that implementers list play speed values as close as possible to the actual speed used in the distribution of the content. Specifically, implementers need to avoid the use of rounding. For example, a speed value of 1.8x should be represented as 9/5 and not rounded to 2. UPnP control points use the speed value in ps-param to compute elapsed time. Rounding errors accumulate quickly resulting in differences between the actual playtime and the value computed by the UPnP control point. The computed value is typically displayed in the control point UI.

7.4.1.3.21.2

[GUIDELINE] If the ps-param token appears in a res@protocolInfo or a protocolInfo of a UPnP AV connection, and if the first field of protocolInfo is "http-get", then the Content Source shall be capable of responding (with a Target Response) to requests that indicate a valid and supported play-speed for the content binary. (i.e. The associated HTTP Server Endpoint does not return error code 406 (Not Acceptable), unless other HTTP headers are the cause of the error.)

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	MIU	[44] [45] [46]	2U3HI	

NOTE: When used with a <res> element or UPnP AV connection, the DLNA.ORG_PS parameter identifies the server supports one or more optional server-side play speed operations.

Guideline 7.5.4.3.3.16 describes more information about responding to HTTP requests that use the PlaySpeed.dlna.org HTTP header.

Variants of the [C1] and [C2] examples that involve sources of audio and/or AV content apply to this guideline.

7.4.1.3.21.3

[GUIDELINE] In conjunction with the rules defined in 7.4.1.3.21.1 and 7.4.1.3.21.2, the fourth field of a res@protocolInfo may use the ps-param for non-DLNA media format profiles.

[ATTRIBUTES]

O	A	DMP DMS DMC DMR +DN+ +UP+ +PU+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	KQ9VQ	

NOTE: This guideline permits the use of DLNA.ORG_PS for both DLNA and non-DLNA content. This guideline also permits the parameter to be used for HTTP and RTP and other transports not specified by DLNA.

The following examples apply to this guideline row: [C1], [C2], [C3], [C4], [P1], [P2], [P3], and [P4], except the printing variant for [C1]. Although the printing and download usages do not employ play speeds to achieve the system usage in a normative way, control points that parse 4th field parameters are required to tolerate the presence of the ps-param.

Note that the CDS:CreateObject syntax has restrictions on the 4th field syntax, as described in 7.4.1.7.19.

7.4.1.3.22 MM ci-param (Conversion Indicator Flag)

7.4.1.3.22.1

[GUIDELINE] The syntax definition of ci-param shall be as follows:

- ci-param = ci-param-delim "DLNA.ORG_CI=" ci-value
- ci-param-delim = ";"
- ci-value = Boolean
- Boolean = "1" | "0"

If the context of the protocolInfo involves a content binary that is converted from a different content binary, then ci-value is "1". Otherwise, the ci-value is "0". (See 7.4.1.3.12 for determining an appropriate context.).

[ATTRIBUTES]

M	A	DMP DMS DMC DMR DMPr +DN+ +UP+ +PU+ +PR1+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	RVR5L	E

NOTE: The ci-param is a 'conversion indication parameter'.

An MSCP uses this parameter to select the most relevant resource from the available resources of a CDS object. If "1" is specified for this parameter value, then the resource is converted from a different content binary. Converted content usually has equal or worse quality.

Examples of conversion include transcoding, system layer conversion, timestamps (e.g. TTS, PCR, PTS), scaling, and decoding.

This guideline also applies in upload AnyContainer and optional content management (OCM) operations. In those cases, a control point is responsible for setting this parameter when it knows the content related to the operation is a converted content binary. This applies for content that intended for a content transfer process, as well operations that involve a URI specified by the control point.

The following examples apply to this guideline: [C1], [C2], [C3], [C4], [C5], [P1], [P2], [P3], [P4], and [P5].

7.4.1.3.22.2

[Guideline] In conjunction with the rules defined in 7.4.1.3.22.1, the fourth field of a res@protocolInfo may use ci-param for the following scenarios.

- The content is conformant to a DLNA media format profile.
- The content is conformant to a non-DLNA media format profile.
- The first field of protocolInfo is "http-get" or "rtsp-rtp-udp "
- The first field of protocolInfo is not "http-get" or "rtsp-rtp-udp "

[Attributes]

O	A	DMP DMS DMC DMR DMPr +DN+ +UP+ +PU+ +PR1+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	VR5L6

NOTE: This guideline permits the use of DLNA.ORG_CI for both DLNA and non-DLNA content. This guideline also permits the parameter to be used for HTTP and RTP and other transports not specified by DLNA.

The following examples apply to this guideline: [C1], [C2], [C3], [C4], [C5], [P1], [P2], [P3], [P4], and [P5].

7.4.1.3.22.3

[Guideline] If the UPnP MediaServer knows that the content of a <res> element is a converted content binary, then the MediaServer should use the ci-param in the protocolInfo value.

[Attributes]

S	A	DMS	M-DMS	MIU	[46]	Q9VQY

NOTE: The ci-param is not mandatory, but its use is strongly encouraged, especially for converted content.

How a Content Source knows if content is converted is out of scope for the guidelines. In some cases, the Content Source knows because it is the entity that actually converts the content. In other cases, the Content Source might know because the Content Source was informed in an implementation-specific manner.

This guideline applies to MediaServers because of examples [C1] and [C2].

7.4.1.3.22.4

[Guideline] The ci-param may be omitted.

[Attributes]

O	A	DMP DMS DMC DMR DMPr +DN+ +UP+ +PU+ +PR1+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[46]	U3HIZ

NOTE: This following examples apply to this guideline: [C1], [C2], [C3], [C4], [C5], [P1], [P2], [P3], [P4], and [P5].

7.4.1.3.22.5

[Guideline] If a protocolInfo value omits the ci-param, then UPnP MediaServer control points shall infer that the associated content is not converted content.

[Attributes]

M	A	DMP DMC DMR DMPr +DN+ +UP+ +PU+ +PR2+	M-DMP M-DMD M-DMC M-DMU	MIU	[46]	D3U5P

NOTE: The following examples apply to this guideline: [P1], [P2], [P3], [P4].

7.4.1.3.23 MM flags-param (Flags Parameter)

7.4.1.3.23.1

[Guideline] The syntax definition of the flags-param shall be as follows

- flags-param = flags-param-delim "DLNA.ORG_FLAGS=" flags-value
- flags-param-delim = ";"
- flags-value = primary-flags reserved-data
- primary-flags = 8 hexdigit
- reserved-data = 24 reserved-hexdigit
- hexdigit = <hexadecimal digit: "0"-"9", "A"-"F", "a"-"f">
- reserved-hexdigit = "0"

If the protocolInfo value omits the flags-param, then the default meaning for individual flags and values embedded in flags-param is determined by default value policies or is unknown (in the case where default values are not defined). As new flags and values are defined for flags-param, those guidelines will clarify the meaning for when flags-param is omitted or when "0" is used.

Example

- DLNA.ORG_FLAGS=03100000000000000000000000000000.

[Attributes]

M	A	DMP DMS DMC DMR DMPr +DN+ +UP+ +PU+ +PR1+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	5UX6U

NOTE: Many DLNA binary-value flags that belong in the fourth field are encapsulated in this parameter. This helps reduce the length of the 4th field as the number of binary-value parameters increases in the future. In a simple usage, a single bit in the binary representation of flags-value maps to a single binary-value parameter. In some cases, DLNA might choose to define that a series of bits represents a small integer.

The following examples apply to this guideline: [C1], [C2], [C3], [C4], [P1], [P2], [P3], and [P4].

7.4.1.3.23.2

[Guideline] The primary-flags token shall be exactly 8 hexadecimal digits and it shall represent a value composed of 32 binary bits. Each bit shall represent a binary flag. The least significant bit corresponds to bit-0 and the most significant bit corresponds to bit-31 (e.g. 10000000000000000000000000000000b = 0x80000000 where bit-31 is the only bit set to true)

The bit mapping of primary-flags shall be as follows:

- Bit-31: sp-flag (Sender Paced Flag)
  - Applies to all DLNA transport protocols.
  - If the flags-param is omitted or the dlna-v1.5-flag is false, then this flag shall have a value of unknown because sender-paced content and non sender paced content is permitted by previous versions of the DLNA guidelines. Content Receivers that fail to transfer and/or render content because they use transport flow control mechanisms are in violation of this guideline.
  - See the following for more information.
    - 7.4.1.3.27 MM sp-flag (Sender Paced Flag)
- Bit-30: lop-npt (Limited Operations Flags: Time-Based Seek)
  - Applies to all DLNA transport protocols.
  - If the flags-param is omitted or the dlna-v1.5-flag is false, then this flag shall have an inferred value of false.
  - See the following for more information.
    - 7.4.1.3.28 MM lop-npt, lop-bytes and lop-cleartextbytes (Limited Operations Flags): Common
- Bit-29: lop-bytes (Limited Operations Flags: Byte-Based Seek)
  - Applies to all DLNA transport protocols.
  - If the flags-param is omitted or the dlna-v1.5-flag is false, then this flag shall have an inferred value of false.
  - See the following for more information.
    - 7.4.1.3.28 MM lop-npt, lop-bytes and lop-cleartextbytes (Limited Operations Flags): Common
- Bit-28: playcontainer-param (DLNA PlayContainer Flag)
  - Applies to all DLNA transport protocols.
  - If the flags-param is omitted or the dlna-v1.5-flag is false, then this flag shall have an inferred value of false because this flag applies only to DMR devices and the DLNA PlayContainer URI operation is optional for DMR devices.
  - See the following for more information.
    - 7.4.1.3.31 MM playcontainer-param (DLNA PlayContainer Flag)
    - 7.4.1.4.25 MM Rendering Media Collection Files
    - 7.4.1.4.28 MM DLNA PlayContainer URI
    - 7.4.1.4.29 MM Control Point Rules for DLNA PlayContainer URI
- Bit 27: s_0-increasing (UCDAM s_0 Increasing Flag)
  - Applies to all DLNA transport protocols.
  - If the flags-param is omitted or the dlna-v1.5-flag is false, then the s_0-increasing flag shall have an inferred value of unknown. The previous guidelines do permit s_0-increasing behavior, but the previous version of the DLNA guidelines (i.e. v1.0) do not define normative rules for using the Seek media operation (or the Range or TimeSeekRange.dlna.org headers) with such content.
  - See the following for more information.
    - 7.4.1.3.32 MM so-increasing (UCDAM so Increasing Flag)
- Bit 26: s_N-increasing (UCDAM s_N Increasing Flag)
  - Applies to all DLNA transport protocols.

- If the flags-param is omitted or the dlna-v1.5-flag is false, then this flag shall have an inferred value of unknown because previous versions of the DLNA guidelines permit content that grows with time or has a fixed ending.
- See the following for more information.
  - 7.4.1.3.33 MM s_N-increasing (UCDAM s_N Increasing Flag)

- Bit-25: rtsp-pause (Pause media operation support for RTP Serving Endpoints)
  - Applies only to RTP Media Transport
  - If the flags-param is omitted or the dlna-v1.5-flag is false, then this flag shall have an inferred value of false because previous versions of the DLNA guidelines do not support the RTP Media Transport.
- Bit 24: tm-s (Streaming Mode Flag)
  - Applies to all DLNA transport protocols.
  - AV and Audio Media Class content shall set at least the tm-s flag equal to true.
  - If the flags-param is omitted or the dlna-v1.5-flag is false, then this flag shall have an inferred value of true only for Audio-only and AV content. For all other content, the inferred value is false.
  - See the following for more information.
    - 7.4.1.3.34 MM tm-s (Streaming Mode Transfer Flag)
- Bit 23: tm-i (Interactive Mode Flag)
  - Applies to all DLNA transport protocols.
  - Image Media Class content shall set at least the tm-i flag equal to true.
  - If the flags-param is omitted or the dlna-v1.5-flag is false, then this flag shall have an inferred value of true only for Image content, XHTML Print documents, and media collection files. For all other content, the inferred value shall be false.
  - See the following for more information.
    - 7.4.1.3.35 MM tm-i (Interactive Mode Transfer Flag)
- Bit 22: tm-b (Background Mode Flag)
  - Applies to all DLNA transport protocols, except RTP.
  - If the flags-param is omitted or the dlna-v1.5-flag is false, then this flag shall have an unknown value.
  - See the following for more information.
    - 7.4.1.3.36 MM tm-b (Background Mode Transfer Flag)
- Bit 21: http-stalling (HTTP Connection Stalling Flag)
  - Applies only to the HTTP Media Transport.
  - If the flags-param is omitted, then this flag shall have an inferred value of false.
  - See the following for more information.
    - 7.4.1.3.37 MM http-stalling (HTTP Connection Stalling Flag)
    - 7.4.1.4.6 MM DIDL-Lite Multiple Res: Thumbnails
- Bit 20: dlna-v1.5-flag (DLNA v1.5 versioning flag)
  - Applies to all DLNA transport protocols.
  - If the flags-param is omitted, then this flag shall have an inferred value of false.
  - See the following for more information.
    - 7.4.1.3.24 MM dlna-v1.5-flag (DLNAv1.5 Version Flag)
- Bit 16: LP-flag (Link Protected Content Flag)

  - Applies to all DLNA transport protocols.
  - If the flags-param is omitted then this flag shall have an inferred value of false.
  - If the cleartextbyteseek-full flag or the lop-cleartextbytes-flag are set then this flag shall be set to true
  - See the following for more information.
    - 7.5.3.5 in [57]
- Bit 15: cleartextbyteseek-full flag
  - (Cleartext Byte Full Data Seek Flag)
  - Applies to all DLNA transport protocols.
  - If the content described by this protocolInfo does not use a Link Protection system (i.e. the LP-flag is false or omitted), the cleartextbyteseek-full flag shall be omitted or set to false.
  - If the flags-param is omitted then the cleartextbyteseek-full flag shall have an inferred value of false.
  - See the following for more information.
    - 7.5.3.6 in [57]:
      (Byte based full seek data availability with the Cleartext Byte Seek Request Header)
- Bit 14: lop-cleartextbytes flag
  - (Cleartext Limited Data Seek Flag)
  - Applies to all DLNA transport protocols.
  - If the content described by this protocolInfo does not use a Link Protection System (i.e. the LP-flag is false or omitted), the lop-cleartextbytes flag shall be omitted or set to false.
  - If the dlna-v1.5 flag is false, then the lop-cleartextbytes flag shall have a value of false.
  - If the flags-param is omitted then the lop-cleartextbytes flag shall have an inferred value of false.
  - See the following for more information.
    - 7.5.3.7 in [57]
      (Byte based limited seek data availability with Cleartext Byte Seek Request Header)

All other bits in primary-flags are reserved for future use and shall have a value of false.

[ATTRIBUTES]

M	C	DMP DMS DMC DMR DMPr +DN+ +UP+ +PU+ +PR1+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	3WJUU

NOTE: The first 8 hexadecimal digits represent 32 binary flags. DLNA defines meaning for some of these bits. Other bits are reserved for future definition and are required to have a value of false at this time.

Note that the usages of defined bits are defined in other DLNA guidelines.

When a protocolInfo represents the capabilities of a content binary, the bits are intended to be a representation of the applicability to the content binary, not on the current conditions of the network or the server's ability to stream data at the current time. When a stream is attempted on a content binary with the tm-s flag set, the DMS is still free to return an error that the stream cannot be completed at the current time due to internal conditions.

The following examples apply to this guideline: [C1], [C2], [C3], [C4], [P1], [P2], [P3], and [P4]. In some cases, the flags-param will provide information that is of no use to certain endpoints. However, endpoints that parse the 4th field are required to tolerate the presence of the flags-param.

7.4.1.3.23.3

[GUIDELINE] The reserved-data shall be exactly 24 hexadecimal digits. These hexadecimal digits are reserved for future use and shall have a value "0".

[ATTRIBUTES]

M	C	DMP DMS DMC DMR +DMPr +DN+ +UP+ +PU+ +PR1+ +PR2	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	YPSTR	

NOTE: The first 8 hexadecimal digits are used for primary-flags and the rest are reserved and undefined at this time.

When DLNA defines new normative flags and/or parameters for the 4th field, those flags and parameters are expected to be defined here. The alternative of declaring a new 4th field parameter is strongly discouraged for future authors of DLNA guidelines.

In general, definitions of new flags or parameters (that will occupy the current space allocated for reserved-bytes) need to define a syntax and semantics for DMS, DMR, DMC, and Push Controllers. Furthermore parameter values that have short hexadecimal representations (such as new binary flags) preferably occupy hexadecimal digits that are more significant while parameter values that have longer hexadecimal representations occupy the less significant hexadecimal digits.

The following examples apply to this guideline: [C1], [C2], [C3], [C4], [P1], [P2], [P3], and [P4].

7.4.1.3.23.4

[GUIDELINE] UPnP AV MediaServer control points shall be tolerant of reserved-data tokens that do not have "0" value hexadecimal digits.

[ATTRIBUTES]

M	C	DMP DMC DMR DMPr +DN+ +UP+ +PU+ +PR2+	M-DMP M-DMD M-DMC M-DMU	MIU	[45] [46]	9OTHY	

NOTE: The following examples apply to this guideline: [P1], [P2], [P3], and [P4].

7.4.1.3.24 MM dlna-v1.5-flag (DLNAv1.5 Version Flag)

[GUIDELINE] If the dlna-v1.5-flag of the primary-flags token is true, then it shall mean the following.

- Bits [31,21] and Bits [16,14] (inclusive) of the primary-flags token are valid for use (i.e. those bits are valid for use)
- Bits [19,17] and Bits [13,0] of the primary-flags token have undefined values.

If the dlna-v1.5-flag of the primary-flags token is false, then it shall mean the following.

- Only Bit-21 of the primary-flags token is defined for use
- All other bits in the primary-flags token have inferred values as described in 7.4.1.3.23.2 (MM flags-param (Flags Parameter)).

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	n/a	SOWQ7	C

NOTE: Bits [31,22] and Bits [16,14] are not defined for previous versions of the DLNA guidelines.

Bit-21 is the only bit defined in an erratum for the DLNA v1.0 guidelines.

7.4.1.3.25 MM maxsp-param (Maximum RTSP Speed header value)

[GUIDELINE] The definition of maxsp-param shall be as follows:

- maxsp-param = maxsp-param-delim "DLNA.ORG_MAXSP=" maxsp-param-value
- maxsp-param-delim = ";"
- maxsp-param-value = 1*DIGIT ["." *DIGIT]

The value of maxsp-param-value shall be greater than or equal to 1.

If maxsp-param is specified, then the RTP Serving Endpoint shall support the Speed header in a RTSP PLAY request if the value of the "Speed" header is less than or equal to the attribute value of maxsp-param. The RTSP "Speed" header is defined in RFC-2326 ([27]), subclause 12.35.

[ATTRIBUTES]

M	A	DMP DMS DMC DMR +DN+ +UP+ +PU+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[27] [44] [45] [46]	W7RAO	

NOTE: Excluding the variants that involve printing and download, the following examples apply to this guideline: [C1], [C2], [C3], [C4], [P1], [P2], [P3], and [P4]. Endpoints that parse 4th field parameters need to tolerate the presence of this parameter.

7.4.1.3.26 MM other-param (Vendor-defined 4th field Parameters)

7.4.1.3.26.1

[GUIDELINE] The definition of other-param is as follows:

- other-param = other-param-delim IANA-name "_" other-param-name "=" other-param-value
- other-param-delim = ";"
- IANA-name = <IANA-registered name, with top level domain (e.g. .net, .org, .com)>
- other-param-name = *<"a"-"z", "A"-"Z", "0"-"9">
- other-param-value = *<"a"-"z", "A"-"Z", "0"-"9", "_", ",", "+", "-">.

[ATTRIBUTES]

M	A	DMP DMS DMC DMR DMPr +DN+ +UP+ +PU+ +PR1+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	8SUNZ	

NOTE: This defines the syntax for vendor extensions in the fourth field of protocolInfo values.

The following examples apply to this guideline: [C1], [C2], [C3], [C4], [P1], [P2], [P3], and [P4].

7.4.1.3.26.2

[GUIDELINE] Vendors may use other-param for vendor-specific parameters in the fourth field of a protocolInfo value.

[ATTRIBUTES]

O	A	DMP DMS DMC DMR DMPr +DN+ +UP+ +PU+ +PR1+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	XQQ6Y	

NOTE: This guideline permits the use of vendor extensions in the fourth field.

The following examples apply to this guideline: [C1], [C2], [C3], [C4], [C5], [P1], [P2], [P3], and [P4].

7.4.1.3.26.3

[GUIDELINE] A UPnP AV MediaServer that receives a CDS:CreateObject request to create a <res> element with a res@protocolInfo that has optional and/or vendor-defined 4th field parameters and if the MediaServer returns a success response, then the MediaServer shall omit the unsupported 4th field parameters from the created <res> element.

[ATTRIBUTES]

M	A	DMS	M-DMS	MIU	[45] [46]	QQ6YX

NOTE: DMS devices are not required to interpret, understand, store, or maintain vendor-defined parameters in the 4th field.

This guideline is an extension of guideline 7.4.1.7.24.3, which requires a DMS to return only the supported metadata properties in a success response.

The following examples apply to this guideline: [P5]. As a corollary, control points involved in example [C5] are expected to tolerate responses that omit unsupported 4th field parameters.

7.4.1.3.27 MM sp-flag (Sender Paced Flag)

7.4.1.3.27.1

[GUIDELINE] The sp-flag (Sender Paced Flag) indicates if the Content Source will act as the Clock Source, for the context of the protocolInfo.

- False = The Content Source is not the Content Clock Source
- True = The Content Source is the Content Clock Source

[ATTRIBUTES]

M	C	DMP DMS DMC DMR +DN+ +UP+ +PU+ +PR2+	M-DMP M-DMD M-DMS M-DMC M-DMU	MIU	[44] [45] [46]	SUNZV

NOTE: The Sender Paced Flag provides a way for the Content Source to indicate that it will send packets at the rate of the normal Clock Source.

In normal HTTP operation the Content Receiver endpoint is the source for the Playback Clock which controls the pace of the rendering. The Content Receiver endpoint uses TCP flow control to match the pace of the transfer of content to the pace of the playback. In some cases, the Content Source might be the Content Clock Source, such as in the case of live broadcast content. This means that if the actual throughput (including any transmission delays caused by additional transmission loads on the network) is not sufficient for the Content Clock Source, then the Content Source will take steps to ensure that the transmitted content binary matches the indicated media format profile, but the bitstream can show discontinuities through things like dropped frames.

Likewise, in RTP the Serving Endpoint is typically the Clock Source and controls the rate at which the content is sent to the network.

The following examples apply to this guideline: [C1], [C2], [C3], [C4], [P1], [P2], [P3], and [P4]. DMPr and +PR1+ are excluded from this guideline because images do not have a Clock Source. However +PR2+ is required to tolerate the sp-flag when browsing a MediaServer.

7.4.1.3.27.2

[GUIDELINE] In the context of SourceProtocolInfo values obtained from CMS:GetProtocolInfo for a UPnP AV MediaServer, the sp-flag shall mean the following.

- False = The Content Source is never the Content Clock Source for a protocolInfo when there is a "match by protocolInfo format".
- True = The Content Source is capable of being the Content Clock Source for a protocolInfo when there is a "match by protocolInfo format".

[ATTRIBUTES]

M	C	DMS	M-DMS	MIU	[45] [46]	7RAO4	

NOTE: Variants of the [C1] and [C2] examples that involve MediaServers apply to this guideline.

7.4.1.3.27.3

[GUIDELINE] In the context of SinkProtocolInfo values obtained from CMS:GetProtocolInfo for a UPnP AV MediaRenderer, the sp-flag shall always be true or the flags-param shall be omitted.

[ATTRIBUTES]

M	L	DMR	n/a	MIU	[44] [45]	OWQ7V	

NOTE: Like DMP devices, DMR devices are required to support rendering of content, regardless of whether the Content Clock Source is determined by the Content Source.

7.4.1.3.28 MM lop-npt, lop-bytes and lop-cleartextbytes (Limited Operations Flags): Common

7.4.1.3.28.1

[GUIDELINE] If the flags-param is present and any of the the lop-npt, lop-bytes, or lop-cleartextbytes bits are true, then the "Limited Random Access Data Availability" model shall be the data access model that applies in the context of the protocolInfo value and the op-param shall be omitted from the 4th field of the protocolInfo value. In addition, for link protected content, the cleartextbyteseek-full flag shall be false.

Specifically, this means that the transport operation (that is indicated by the lop-npt or lop-bytes or lop-cleartextbytes) shall be supported for a limited data range for the context of the protocolInfo, as defined in 7.5.4.2.16.

The meaning of the lop-npt bit and the lop-bytes and lop-cleartextbytes flags are described in these guidelines, depending on whether the context is for the HTTP Media Transport or the RTP Media Transport.

- 7.4.1.3.29 MM lop-npt lop-bytes, and lop-cleartextbytes (Limited Operations Flags): HTTP
- 7.4.1.3.30 MM lop-npt and lop-bytes (Limited Operations Flags): RTP

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	[45] [46]	OTHY2	

NOTE: The lop-npt indicates that the transport layer supports random access on a limited range of npt playback positions. Likewise the lop-bytes indicates that the transport layer supports random access on a limited range of byte positions.The lop-cleartextbytes flag indicates that the transport layer supports random access on a limited range of byte positions within the cleartext byte domain.

A fundamental difference between lop-npt/lop-bytes/ lop-cleartextbytes and the op-param is that lop-npt/ lop-bytes / lop-cleartextbytes assumes that s_0 can change, while op-param only permits changes to s_N. For additional information on the assumptions for The "Limited Random Access Data Availability" model, see 7.5.4.2.16.

The following examples apply to this guideline: [C1] and [C2]. +PR1+ does not apply to this guideline because it will never serve images under the "Limited Random Access Data Availability" model.

7.4.1.3.28.2

[GUIDELINE] Content Receiver endpoints and UPnP AV MediaServer control points that attempt to acquire content data from the limited random access data range (defined in 7.5.4.2.16 shall be able to properly request a valid range, even if the limited data range continuously changes with time.

This guideline only applies when "Limited Random Access Data Availability" applies to the scenario.

[ATTRIBUTES]

M	A	DMP DMR +DN+	M-DMP M-DMD	MIU	[44] [45] [46]	PSTRZ

NOTE: Content receiver endpoints and control points cannot assume anything special about the data range because the absolute beginning can change and the content can have no end (i.e. live content). See the following for more general information and examples of how this guideline applies to HTTP.

- 7.5.4.2.16 MT "Limited Random Access Data Availability" Model
- 7.5.4.3.2.20 MT HTTP: Data Range of "Limited Random Access Data Availability", Guidelines 7.5.4.3.2.20.2 and 7.5.4.3.2.20.10

The [P1], [P2], and [P3] examples apply to the Content Receivers (excluding DMPr) that are governed by this guideline. DMPr devices are excluded because images and XHTML print documents are never transferred under the "Limited Random Access Data Availability" model.

7.4.1.3.29 MM lop-npt lop-bytes, and lop-cleartextbytes (Limited Operations Flags): HTTP

[GUIDELINE] If the 4th field is associated with the "http-get" transport protocol, then the lop-npt, lop-bytes and lop-cleartextbytes bits mean the following.

- lop-npt: indicates support of the TimeSeekRange.dlna.org HTTP header for the context of the protocolInfo under the "Limited Random Access Data Availability" model
- lop-bytes: indicates support of the Range HTTP header for the context of the protocolInfo under the "Limited Random Access Data Availability" model
- lop-cleartextbytes: indicates support of the Cleartext Byte Seek Request Header for the context of the protocolInfo under the "Limited Random Access Data Availability" model

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	[33], [45] [46]	WJUUP

NOTE: This guideline defines the lop-npt, lop-cleartextbytes and lop-bytes bits, when HTTP is the transport protocol.

When used with a context involving HTTP, these bits identify if the server supports the TimeSeekRange.dlna.org or Range HTTP headers or the Cleartext Byte Seek Request Header for the associated content binary under the "Limited Random Access Data Availability" model.

For more information on the "Limited Random Access Data Availability" model for HTTP and these HTTP headers, see the following guidelines.

- 7.5.4.2.16 MT "Limited Random Access Data Availability" Model
- 7.5.4.3.2.18 MT HTTP Common Random Access Data Availability Requirements
- 7.5.4.3.2.20 MT HTTP: Data Range of "Limited Random Access Data Availability"

And the following guideline in [57]

- A.5

The following examples apply to this guideline: [C1] and [C2]. +PR1+ does not apply to this guideline because it will never serve images using the "Limited Random Access Data Availability" model.

7.4.1.3.30 MM lop-npt and lop-bytes (Limited Operations Flags): RTP

[GUIDELINE] If the 4th field is associated with the "rtsp-rtp-udp" transport protocol, then the lop-npt and lop-bytes bits mean the following.

- lop-npt: indicates support of the Range header for the context of the protocolInfo under the "Limited Random Access Data Availability" model
- lop-bytes: In scenarios involving the RTP Media Transport, the lop-bytes bit shall be set to false.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	[33], [45] [46]	UX6UW	

NOTE: This guideline defines the lop-npt and lop-bytes bits, when RTP is the transport protocol.

When used with a context involving RTP, these bits identify if the server supports the Range header for the associated content binary under the "Limited Random Access Data Availability" model.

For more information on the "Limited Random Access Data Availability" model for RTP and these RTP headers, see the following guidelines.

- 7.5.4.2.16 MT "Limited Random Access Data Availability" Model
- 7.5.4.4.6.2.53 MT RTP Current Limited Data Range indication

The following examples apply to this guideline: [C1] and [C2]. +PR1+ does not apply to this guideline because it will never serve images using the RTP Media Transport.

7.4.1.3.31 MM playcontainer-param (DLNA PlayContainer Flag)

7.4.1.3.31.1

[GUIDELINE] The playcontainer-param flag indicates support for a DLNA PlayContainer URI operation. If the flag is true for a protocolInfo, then it means that the UPnP AV MediaRenderer can play that type of content in a DLNA PlayContainer URI operation.

The playcontainer-param flag shall be false when the context of the protocolInfo involves a media collection binaries (e.g. DIDL_S and DIDL_V playlist files). Furthermore, when performing a DLNA PlayContainer URI, the MediaRenderer shall not render media collection binaries when traversing the CDS hierarchy. Note that this restriction against playing media collection binaries applies only to media collection binaries defined by the DLNA guidelines.

[ATTRIBUTES]

M	A	DMR DMC +PU+	M-DMC	MIU	[44] [45] [46]	3U5PA	

NOTE: A DLNA PlayContainer URI allows a control point to instruct a DMR to browse a DMS and play content from it.

The playcontainer-param is used on a per-profile basis. For example, if the protocolInfo for "http-get" and "MPEG2_PS_NTSC" has playcontainer-param set to true, then MPEG2_PS_NTSC content will be played in playcontainer operation.

Example [C4] also applies to this guideline.

This also guideline applies to DMR devices because they can expose protocolInfo that has the playcontainer-param set to true through CMS:GetProtocolInfo.

This guideline also applies to control points that invoke CMS:GetProtocolInfo on a DMR because they have to parse 4th field values.

Since the res@dlna:trackTotal attribute is not required, there is not a consistent way to represent the individual tracks of media collection binaries. Furthermore, DLNA has no interoperability guidelines for navigating the tracks within a media collection binary. Therefore, these guidelines prohibit playback of media collection binaries until a future set of DLNA guidelines can adequately address these issues.

7.4.1.3.31.2

[GUIDELINE] In the context of the Source argument's protocolInfo values obtained from CMS:GetProtocolInfo for a UPnP AV MediaServer, the playcontainer-param flag shall always be false if the flags-param is included for a protocolInfo value.

Likewise, 4th field values provided by a Content Source shall set the playcontainer-param flag to false if the flags-param is included in the protocolInfo value.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+	M-DMS	MIU	[45]	3HIZ9

NOTE: The DMR device class is the only device class that sets the playcontainer-param to true.

The following examples apply to this guideline: [C1] and [C2].

7.4.1.3.32 MM s_0-increasing (UCDAM s_0 Increasing Flag)

7.4.1.3.32.1

[GUIDELINE] The s_0-increasing (UCDAM s_0 Increasing Flag) indicates if the UCDAM s_0 boundary is increasing.

- True = The s_0 data boundary increases with time.
- False = The s_0 data boundary is fixed.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	[44] [45] [46]	9VQYF

NOTE: If true, then the content does not have a fixed beginning. Otherwise, the content does have a fixed beginning (i.e. npt=0 and byte-pos=0 map to the beginning). Note that the s0 data boundary can reset, as described in the comments 7.5.4.2.16 guideline 7.5.4.2.16.3.

The [C1] and [C2] examples apply to this guideline, with exception of those involving +PR1+ because images do not have beginnings that change with time.

7.4.1.3.32.2

[GUIDELINE] If the s_0-increasing flag is true then the following shall apply to the context of the protocolInfo.

- The op-param shall be omitted.
- If lop-npt and lop-bytes are both false, then following shall also apply.
- The s_0 data boundary shall map to a beginning that is not static.
- The data range of [s_0, s_N] shall map to the npt range of [npt-start-time, npt-last-time] and the byte range of [first-byte-pos, last-byte-pos], where npt-start-time and npt-last time are in units of npt and first-byte-pos and last-byte-pos are in units of bytes.
- There exists a "live position" that shall be equal to the s_N data boundary.
- If the s_N data boundary is changing with time, then the "live position" shall shift forward in real-time.
- If the Content Source receives a transport layer request that is not a random access request (e.g. HTTP request that omits Range and TimeSeekRange.dlna.org) then the Content Source shall respond with content data from the "live point".
- If either lop-npt or lop-bytes is true, then the "Limited Random Access Data Availability" model shall apply. See 7.5.4.2.16 for more information.

The rules in this guideline shall apply even in scenarios where the transport server does not support random access requests (e.g. HTTP requests with Range or TimeSeekRange.dlna.org) for a content binary.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	[44] [45] [46]	R5L6Y

NOTE: When s_0-increasing is true, then the content binary has a beginning that can change. Since it is possible to set the s_0-increasing flag to true and not support random access requests, it is necessary for this guideline to be applied in all scenarios.

For HTTP, this means that omitting the Range and TimeSeekRange.dlna.org headers in an HTTP GET request results in the HTTP Server Endpoint returning content data bytes from a "live position" (as described in 7.5.4.3.2.20.9). Furthermore npt=0 or byte-pos=0 has no meaning (as described by 7.5.4.3.2.20.16).

If lop-npt or lop-bytes is true, then the access model is governed by what is returned by availableSeekRange.dlna.org (as described in 7.5.4.3.2.20).

The [C1] and [C2] examples apply to this guideline, with exception of those involving +PR1+ because images do not have beginnings that change with time.

7.4.1.3.32.3

[GUIDELINE] If the s_0-increasing flag is false, then the following shall apply to the context of the protocolInfo.

- The s_0 data boundary shall map to a fixed and non-changing beginning.
- The data range of [s_0, s_N] shall occupy an npt range of [0, npt-last-time] and a byte range of [0, last-byte-pos], where npt-last-time is in units of npt and last-byte-pos is in units of bytes.
- The content binary's zero position (i.e. npt-time=0 and byte-pos=0) shall map to the UCDAM's data position of s_0.
- The last-byte-pos and npt-last-time shall map to the UCDAM's s_N data position and the s_N data boundary shall map to the end of the available content data.

This guideline shall apply even in scenarios where the transport server does not support random access requests (e.g. HTTP requests with Range or TimeSeekRange.dlna.org) for a content binary.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	n/a	5L6YK	

NOTE: When s_0-increasing is false, then the content binary has a fixed beginning. Since it is possible to set the s_0-increasing flag to false and not support random access requests, it is necessary for this guideline to be applied in all scenarios.

For HTTP, this guideline means that HTTP requests that omit the Range and TimeSeekRange.dlna.org headers results in the HTTP Server Endpoint returning content data from the absolute beginning of the content (as described by 7.5.4.3.2.19).

Note that these assumptions and the required behavior are consistent with assumptions of previous versions of the DLNA guidelines.

Note that s_0-increasing=false permits mutually exclusive use of either the op-param or the lop-npt/lop-bytes with values of true, while s_0-increasing=true prohibits use of the op-param but allows use of lop-npt and lop-bytes.

The [C1] and [C2] examples apply to this guideline, with exception of those involving +PR1+ because images do not have beginnings that change with time.

7.4.1.3.32.4

[GUIDELINE] In the context of Sink argument's protocolInfo values obtained from CMS:GetProtocolInfo for a UPnP AV MediaRenderer, the s_0-increasing flag shall always be true or the flags-param shall be omitted.

[ATTRIBUTES]

M	L	DMR	n/a	MIU	[44] [45] [46]	VQYFH	

NOTE: Like DMP devices, DMR devices are required to support normal playback rendering of content, regardless of the server's buffering model.

7.4.1.3.33 MM s_N-increasing (UCDAM s_N Increasing Flag)

7.4.1.3.33.1

[GUIDELINE] The s_N-increasing (UCDAM s_N Increasing Flag) indicates if the UCDAM s_N boundary is increasing.

- True = The s_N data boundary increases with time.
- False = The s_N data boundary is fixed.

This flag applies regardless of whether the "Full Random Access Data Availability" or "Limited Random Access Data Availability" models is being used.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	[44] [45] [46]	HIZ9G

NOTE: If true, then the content does not have a fixed ending. Otherwise, the content has a fixed ending.

In conjunction with 7.4.1.3.32, it is possible to determine whether the server exhibits a growing or sliding buffering model.

The [C1] and [C2] examples apply to this guideline, with exception of those involving +PR1+ because images do not have endings that change with time.

7.4.1.3.33.2

[GUIDELINE] In the context of SinkProtocolInfo values obtained from CMS:GetProtocolInfo for a UPnP AV MediaRenderer, the s_N-increasing flag shall always be true or the flags-param shall be omitted.

[ATTRIBUTES]

M	L	DMR	n/a	n/a	[44] [45]	U5PAG

NOTE: Like DMP devices, DMR devices are required to support normal playback rendering of content, regardless of the server's buffering model.

7.4.1.3.34 MM tm-s (Streaming Mode Transfer Flag)

7.4.1.3.34.1

[GUIDELINE] If the tm-s flag is true, then the associated Media Transport Content Source shall be capable of supporting the Streaming Mode Transfer for the context of protocolInfo.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	n/a	X6UW9

NOTE: See Table 46 for more information about Streaming Mode Transfer.

Note that Content Sources can generate an error response if it does not have the resources to respond at the current time.

The tm-s flag is not equivalent to the sp-flag. When the tm-s flag is true, it means that the Content Source is able to transmit fast enough for immediate rendering. If the sp-flag is false and the sustained throughput is less than what is needed for immediate rendering, then the Content Source will preserve the content binary's bitstream because the Content Source does not act as the Clock Source.

When the sp-flag is also true, it means that the Content Source is also the Clock Source for the content, which means that the Content Source will take steps to ensure that the content binary meets the expectations of the media format profile, but the rendering stream can have discontinuities (such as dropped frames).

The [C1] and [C2] examples apply to this guideline, with exception of those involving +PR1+ because the Streaming Transfer mode has no meaning for images and XHTML print documents.

7.4.1.3.34.2

[Guideline] The tm-s flag shall be set to true for protocolInfo values where the pn-param indicates a media format profile for audio-only or AV media class. Setting the tm-s flag to true for Images, XHTML print documents, or media collection binaries is expressly prohibited.

This requirement does not apply in scenarios where a res@protocolInfo value exists for a <res> element that does not have a URI value. (e.g. a CDS object that was created in an upload AnyContainer operation and has yet to receive the actual content binary)

[Attributes]

M	A	DMS +PU+	M-DMS	MIU	n/a	JUUPX	

NOTE: This ensures backwards compatibility with earlier DMP devices that always assume that audio-only and AV content is available for streaming.

Please see guideline 7.5.4.2.3.2 (MT Transfer Mode Support) for information on the transfer modes that a Content Receivers can specify when issuing requests.

The [C1] and [C2] examples apply to this guideline, with exception of those involving +PR1+ because the Streaming Transfer mode has no meaning for images and XHTML print documents.

7.4.1.3.35 MM tm-i (Interactive Mode Transfer Flag)

7.4.1.3.35.1

[Guideline] If the tm-i flag is true, then the associated Media Transport Content Source shall be capable of supporting the Interactive Mode Transfer for the context of protocolInfo.

[Attributes]

M	A	DMS +PR1+ +PU+	M-DMS	MIU	n/a	STRZQ	

NOTE: See Table 46 for more information about Interactive Mode Transfer.

Note that Content Sources can generate an error response if it does not have the resources to respond at the current time.

The [C1] and [C2] examples apply to this guideline.

7.4.1.3.35.2

[Guideline] The tm-i flag shall be set to true for protocolInfo values where the pn-param indicates a media format profile for the Image media class, XHTML print documents, or a media collection binary. Setting the tm-i flag to true for Audio-only or AV content is expressly prohibited.

This requirement does not apply in scenarios where a res@protocolInfo value exists for a <res> element that does not have a URI value. (e.g. a CDS object that was created in an upload AnyContainer operation and has yet to receive the actual content binary)

[Attributes]

M	A	DMS +PR1+ +PU+	M-DMS	MIU	n/a	THY23	

NOTE: This ensures backwards compatibility with earlier DMP devices that always assume that image content is available for immediate rendering.

Please see guideline 7.5.4.2.3.2 (MT Transfer Mode Support) for information on the transfer modes that a Content Receivers can specify when issuing requests.

The [C1] and [C2] examples apply to this guideline.

7.4.1.3.36 MM tm-b (Background Mode Transfer Flag)

[GUIDELINE] If the tm-b flag is true, then the associated HTTP server shall be capable of supporting the Background Mode Transfer for the context of the protocolInfo.

In the context of a UPnP AV Connection, a res@protocolInfo value, or contentFeatures.dlna.org value, the following restrictions shall also apply:

- If the http-stalling flag is true, then tm-b flag shall be set to true.
- If the sp-flag is true, then tm-b flag shall be false.

[ATTRIBUTES]

M	A	DMS +PR1+ +PU+	M-DMS	MIU	n/a	WQ7V6	

NOTE: See Table 46 for more information about Background Mode Transfer.

Note that Content Sources can generate an error response if it does not have the resources to respond at the current time.

Unlike the tm-s and tm-i flags, the tm-b flag can be used with all media classes.

Note that a server that supports the http-stalling flag will also support the tm-b flag because a device that supports indefinite stalling implicitly supports lower transmission throughputs that can result from actively managing TCP flow control for a Background transfer. The converse is not true becase the ability to support a Background transfer does not necessarily imply the ability to support indefinite stalling via TCP flow control.

Also note that a Background transfer cannot be used in conjunction with server-paced content. Lower transmission throughputs resulting from a Background transfer can cause the server's buffer to overflow. Content Receivers that want to download content that have sp-flag=true need to use the streaming download media operation.

Please see guideline 7.5.4.2.3.2 (MT Transfer Mode Support) for information on the transfer modes that a Content Receivers can specify when issuing requests.

The [C1] and [C2] examples apply to this guideline.

7.4.1.3.37 MM http-stalling (HTTP Connection Stalling Flag)

[GUIDELINE] If the http-stalling flag is true, then the associated HTTP server shall be capable of supporting the Connection Stalling method for the Pause and Pause-Release media operations on the content binary and in addition the sp-flag shall be false.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	n/a	RAO4V	

NOTE: The Connection Stalling is a mechanism where a Content Receiver and a Content Source cooperatively use standard TCP flow control to temporarily pause the transmission of data.

HTTP Server Endpoints are not to misinterpret HTTP-level transport inactivity as a symptom of a TCP disconnect because a properly stalled HTTP Client Endpoint will use standard TCP flow control to keep the TCP connection alive. HTTP Server Endpoints also need to be careful to not overflow their local network buffers when theConnection Stalling method is being used.

This [C1] and [C2] examples apply to this guideline when the scenario involves audio or AV content.

7.4.1.3.38 MM UPnP AV Connection Behaviors

7.4.1.3.38.1

[GUIDELINE] A UPnP AV MediaServer shall have a connection with a ConnectionID "0" to represent all transport layer connections that cannot be mapped to a particular transport layer connection.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[45] [49]	UNZVD	

NOTE: A UPnP AV MediaServer that implements this type of behavior uses ConnectionID "0" to represent one or more transport layer connections. Specifically, UPnP AV MediaServers are required to have at least one UPnP AV connection with ConnectionID of "0" to represent an unknown number of connections (zero or more) to Content Receivers.

7.4.1.3.38.2

[GUIDELINE] A UPnP AV MediaRenderer shall have a connection with a ConnectionID value of "0" to represent the default connection whose access is always available to any UPnP AV MediaRenderer control point in the network.

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[45] [49]	RBWYA

NOTE: UPnP AV MediaRenderers maintain a default connection that can be accessed at any time by any networked UPnP AV MediaRenderer control point (DMC, M-DMC, +PU+). A ConnectionID value of 0 identifies this default connection.

7.4.1.3.39 MM Context of ConnectionID=0

7.4.1.3.39.1

[GUIDELINE] UPnP AV MediaServer control points shall not rely on the accuracy of the *ProtocolInfo* and *PeerConnectionManager* output parameters from a CMS:GetCurrentConnectionInfo request to a UPnP AV MediaServer with a value of 0 for the *ConnectionID* input.

[ATTRIBUTES]

M	A	DMP DMC +DN+ +UP+	M-DMP M-DMC M-DMD M-DMU	MIU	[45]	Q6YXR

NOTE: In the case of the UPnP AV connection "0", the information is assumed to be inaccurate because the context of ConnectionID="0" represents numerous requests for different content. For example, the same ConnectionID of value "0" can be used for simultaneously serving an image and an audio stream. Each of the media resources served in this scenario will have its own protocolInfo.

7.4.1.3.39.2

[GUIDELINE] When a UPnP AV MediaRenderer responds to a CMS:GetCurrentConnectionInfo request, the value of the *ProtocolInfo* output argument shall contain information that corresponds to the URI in the AVT.CurrentTrackURI virtual instance state variable of the corresponding AVTransport virtual instance. The corresponding AVTransport virtual instance has an InstanceID equal to the value of the *AVTransportID* output argument of the same response.

If content has been assigned to the AVTransport virtual instance (i.e. the corresponding AVT.CurrentTrackURI virtual instance state variable is not equal to the empty string), the value of the *ProtocolInfo* output argument to the CMS:GetCurrentConnectionInfo request shall at least include the Primary protocolInfo Set that describes the content.

If content has not been assigned yet to the AVTransport virtual instance (i.e. the corresponding AVT.CurrentTrackURI virtual instance state variable equals the empty string), the value of the *ProtocolInfo* output argument to the CMS:GetCurrentConnectionInfo request shall be an empty string.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[45] [49]	R2JS7

NOTE: UPnP AV MediaRenderers always include the proper ProtocolInfo value when responding to CMS:GetCurrentConnectionInfo requests.

7.4.1.3.40 MM UPnP AV Connection ID and Instance ID Assignment Rules

7.4.1.3.40.1

[GUIDELINE] UPnP AV MediaServers and UPnP AV MediaRenderers may include non-zero ConnectionID values in addition to the zero value in the CMS.CurrentConnectionIDs state variable.

[ATTRIBUTES]

O	C	DMS DMR	M-DMS	n/a	[45] [48] [49]	6YXRZ	

NOTE: Typically, UPnP AV MediaServers and UPnP AV MediaRenderers include only the value of 0 when responding to CMS:GetCurrentConnectionIDs. However, devices that implement CMS:PrepareForConnection and/or DLNA BCM guidelines will respond with a list that includes additional non-zero ConnectionID values.

7.4.1.3.40.2

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						NZVDY	

7.4.1.3.40.3

[GUIDELINE] If a UPnP AV MediaServer does not implement the AVTransport service, then it shall return the value of "-1" for the AVTransportID argument in the CMS:GetCurrentConnectionInfo action.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[44] [45] [90]	AO4VF	

NOTE: This is normative per the UPnP AV specifications. The CMS:PrepareForConnection and CMS:GetCurrentConnectionInfo actions return the AVTransport service virtual instance ID value through the *AVTransportID* output argument.

7.4.1.3.40.4

[GUIDELINE] A UPnP AV MediaRenderer shall return the value of "0" in the *AVTransportID* and *RcsID* arguments in response to a CMS:GetCurrentConnectionInfo request where the *ConnectionID* argument has the value of "0".

[ATTRIBUTES]

M	L	DMR	n/a	n/a	[44] [45] [48]	Q7V6G	

NOTE: Guidelines 7.4.1.6.2.1 and 7.4.1.6.3.1 require the "default" virtual instances to be always present in the AVTransport and RenderingControl services. This guideline further requires these "default" virtual instances to be always associated with the "default" connection of the Connection Manager service.

7.4.1.3.40.5

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						HY23H	

7.4.1.3.40.6

[GUIDELINE] UPnP AV MediaServer and UPnP AV MediaRenderer value for the CMS.CurrentConnectionIDs state variable shall be a comma-separated list of all current ConnectionID values.

[ATTRIBUTES]

M	C	DMS DMR	M-DMS	n/a	[45] [48] [49]	B8T5G	

NOTE: Implementing support for the action CMS:GetCurrentConnectionIDs is a requirement for UPnP AV MediaServers and UPnP AV MediaRenderers. By default, one connection is always available. This default connection is identified with a ConnectionID value of 0.

7.4.1.3.40.7

[GUIDELINE] A UPnP AV MediaServer and a UPnP AV MediaRenderer shall respond to CMS:GetCurrentConnnectionIDs requests. The *ConnectionIDs* output argument value shall have the same value as the CMS.CurrentConnectionsIDs state variable.

[ATTRIBUTES]

M	C	DMS DMR	M-DMS	n/a	[45] [48] [49]	I9O4R	

7.4.1.3.40.8

[GUIDELINE] UPnP AV MediaServers and UPnP AV MediaRenderers shall include a ConnectionID value of zero in the CMS.CurrentConnectionIDs state variable.

[ATTRIBUTES]

M	C	DMS DMR	M-DMS	n/a	[45] [48] [49]	BWYAJ	

NOTE: Guideline requirements 7.4.1.3.38.1 and 7.4.1.3.38.2 require that UPnP AV MediaServers and UPnP AV MediaRenderers always have a default connection with ConnectionID "0". This guideline indicates that this value needs to be included in the CMS.CurrentConnectionIDs state variable.

7.4.1.3.40.9

[GUIDELINE] A UPnP AV MediaServer shall return the value of "-1" for the *RcsID* argument in the CMS:GetCurrentConnectionInfo action.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[44] [45] [49]	2JS7H	

This is normative per the UPnP AV specifications. The CMS:PrepareForConnection and CMS:GetCurrentConnectionInfo actions return the RenderingControl service virtual instance ID value through the *RcsID* output argument.

7.4.1.4 MediaServer Requirements

7.4.1.4.1 MM *ObjectID* Usage

7.4.1.4.1.1

[GUIDELINE] UPnP AV MediaServers shall assign a unique object ID for each entry in their ContentDirectory service (CDS). This rule applies to both container and item objects in a CDS metadata hierarchy.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	TRZQR	

NOTE: This is a requirement of the CDS specification. This guideline's scope for uniqueness is for the entire CDS hierarchy.

7.4.1.4.1.2

[GUIDELINE] UPnP AV MediaServers should maintain the object ID value on a persistent basis.

[ATTRIBUTES]

S	L	DMS	M-DMS	n/a	[46]	UUPXM

NOTE: The purpose of this recommendation is to allow control points to implement features like "my favorite content". Although control points cannot assume that object ID values are persisted, this recommendation allows a control point to easily check if a CDS object is still available on the network.

The reason why this is not mandatory is that some embedded devices can have difficulty persisting object ID values.

7.4.1.4.1.3

[GUIDELINE] A UTF-8 encoded object ID shall not exceed 256 bytes in the XML escaped form encoded in UTF-8.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46]	6UW9O

NOTE: Provides a reasonable maximum length for objectID values, which are essential for CDS object declarations.

This guideline only applies to creation of object ID values.

7.4.1.4.1.4

[GUIDELINE] DIDL-Lite documents or fragments that contain one or more CDS objects (i.e. <item> or <container> element) shall have a unique value for each object's @id attribute.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	IZ9GG

NOTE: This is a requirement of the CDS specification. This guideline's scope for uniqueness is limited to the DIDL-Lite document or fragment. For example, a UPnP AV MediaRenderer that reports metadata for a media collection cannot use the same object ID for each of the items in the media collection.

7.4.1.4.2 MM CDS:Browse Unsorted Order

[GUIDELINE] If a UPnP AV MediaServer responds to a CDS:Browse request that specifies *BrowseFlag*=BrowseDirectChildren and an empty *SortCriteria* argument, then the MediaServer shall preserve the indexed order of returned CDS objects for a given *UpdateID* output argument value.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46]	QYFH2

NOTE: This requirement is implied in the CDS specification because the *StartingIndex* and *RequestedCount* input arguments are designed for incrementally browsing a CDS container. The only time when the unsorted order can be different between two CDS:Browse requests will be when the *UpdateID* output argument values are different in the two responses.

7.4.1.4.3 MM DIDL-Lite Multiple Res: Formats

[GUIDELINE] A CDS object (identified through an <item> or <container> element) should use multiple <res> elements (instead of multiple CDS objects) when the <res> elements represent the same content in different media format profiles.

[ATTRIBUTES]

S	R	DMS	M-DMS	n/a	[46]	L6YK5

NOTE: These guidelines encourage a UPnP AV MediaServer device to expose multiple <res> elements for a single CDS object. This allows a UPnP AV MediaServer control point to present a single CDS object to the user, without necessarily presenting each variant of the same content.

This requirement applies when the DMS can determine that content binaries are the same content. The following are some examples of how multiple <res> elements can be used.

- The DMS acquired the content binaries in such a way that it knows that they are the same content.
- The DMS has transcoded locally stored content to other formats.
- The DMS advertises tuner-sourced content in multiple media format profiles.
- The DMS has a single, locally stored file that is advertised with multiple media format profiles, without any conversions.
- The DMS advertise the same video or image content (even when the media format profile is the same) but in different resolutions.

This requirement does not require a DMS to determine whether separate, locally stored files are actually the same content because this is difficult to do computationally and relying on embedded metadata is not always accurate. Furthermore, a DMS is not required to determine that content binaries uploaded through multiple upload AnyContainer or OCM: upload content operations requests are the same content.

This guideline is required because the MediaServer can always determine if it is providing a content transformation.

7.4.1.4.4 MM DIDL-Lite Multiple Res: Transports

[GUIDELINE] A CDS object (identified through an <item> or <container> element) that has the same content available for different media transport protocols shall expose the content through multiple <res> elements.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	6YK54

NOTE: See the comments in 7.4.1.4.3 for more information.

7.4.1.4.5 MM DIDL-Lite Content: Multiple Points of Accessibility

7.4.1.4.5.1

[GUIDELINE] A UPnP AV MediaServer that does not receive the ALLIP value (case sensitive) as part of the Filter argument (in a CDS:Browse or CDS:Search request) shall return only the URIs that are associated with (or treated as or assumed to be routable from) the network interface that received the SOAP request.

URIs with domain names may appear in these types of SOAP responses, according to the rules specified in 7.4.1.3.10.

This guideline applies to URIs in <res>.

This guideline applies to any URI value that uses an IPv4 network address, regardless of whether the content conforms to a DLNA media format profile.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46]	YFH2V

NOTE: These guidelines explain how a UPnP AV MediaServer is to handle the reporting of <res> elements when the UPnP AV MediaServer has multiple network interfaces.

Essentially, the default behavior is that a UPnP AV MediaServer will only return <res> elements where the URIs are known (or assumed to be) routable from the network interface that received the request. However, if a UPnP AV MediaServer control point wants to receive URIs for all network interfaces, then the DMP can specify the ALLIP value as part of the Filter argument. In such a scenario, a UPnP AV MediaServer is obligated to return all of the <res> elements for all of the active network interfaces that the UPnP AV MediaServer uses for media transport.

Because the guidelines language uses the UPnP AV MediaServer of a given UDN, a UPnP AV MediaServer device that uses a different UDN for each network interfaces (equivalent to multiple UPnP AV MediaServers for a single UPnP AV MediaServer device) does not need to return the <res> elements that are accessible on a different network interface (e.g., res element found on a different logical UPnP AV MediaServer).

Appendix C (Informative) UPnP Devices with Multiple Network describes the subtleties of multiple network interfaces and the role of these guidelines in more detail.

7.4.1.4.5.2

[GUIDELINE] A UPnP AV MediaServer that receives the ALLIP value (possibly with other filter values, including the star,*, value) in the Filter argument (of a CDS:Browse or CDS:Search request) shall return all URIs associated with the UPnP AV MediaServer of a given UDN, regardless of whether the URI is thought to be routable from the network interface that received the SOAP request.

A UPnP AV MediaServer shall expose all URI values either through multiple <res> elements (for each CDS object) or multiple CDS objects. Please see guidelines 7.4.1.4.5.3 and 7.4.1.4.5.4 for how all URI values are exposed through multiple <res> elements or multiple CDS objects.

URIs with domain names may appear in these types of SOAP responses, according to the rules specified in 7.4.1.3.10.

This guideline applies to URIs in <res>.

This guideline applies to any URI value that uses an IPv4 network address, regardless of whether the content conforms to a DLNA media format profile.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	Z9GG5	

7.4.1.4.5.3

[GUIDELINE] In conjunction with guideline 7.4.1.4.5.2, a UPnP AV MediaServer that receives the ALLIP value should return CDS objects with multiple <res> elements, such that some of these <res> elements are URI values that point to the same content available on different network interfaces.

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[46]	PAGPN	

NOTE: This guideline allows a UPnP AV MediaServer to report content availability on multiple networks through multiple <res> elements. The presence of multiple <res> elements is still governed by 7.4.1.4.5.2.

7.4.1.4.5.4

[GUIDELINE] A UPnP AV MediaServer that does not receive the ALLIP value (possibly with other filter values, including the asterisk,*, value) may return CDS objects with zero or more <res> elements.

Note that it is generally true that a UPnP AV MediaServer may return CDS objects with zero or more <res> elements.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[46]	UW9OG	

NOTE: Although it is implied that a UPnP AV MediaServer is not required to provide a <res> element for a CDS object, this guideline states it explicitly. This guideline allows implementations that rely on multiple CDS objects (instead of multiple <res> elements to represent different versions of the same content) to comply with 7.4.1.4.5.1.

7.4.1.4.6 MM DIDL-Lite Multiple Res: Thumbnails

7.4.1.4.6.1

[GUIDELINE] If a UPnP AV MediaServer exposes a CDS object with a <upnp:class> designation of object.item.imageItem (or any class derived from it), then the UPnP AV MediaServer should provide a <res> element for the thumbnail resource. (Multiple thumbnail <res> elements are also allowed.)

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[46]	UPXML	

NOTE: UPnP AV MediaServer devices that implement thumbnail support reduce the network load for themselves and for control points that display thumbnails to the user.

7.4.1.4.6.2

[GUIDELINE] If a UPnP AV MediaServer exposes a CDS object with a <upnp:class> designation of object.item.videoItem (or any class derived from it), then the UPnP AV MediaServer should provide a <res> element for the thumbnail resource. (Multiple thumbnail <res> elements are also allowed.)

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[46]	RZQRD	

7.4.1.4.6.3

[GUIDELINE] If a UPnP AV MediaServer exposes thumbnail images for image or video content, then a UPnP AV MediaServer shall provide a thumbnail that conforms to guideline 7.1.7 in [56] media format profile and be declared with the JPEG_TN designation in the fourth field of the res@protocolInfo attribute.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46] [56]	Y23HQ	

NOTE: When thumbnails are provided, the minimal expectation is to provide JPEG thumbnails. However, vendors can also provide additional thumbnails of other formats (such as a PNG thumbnail).

7.4.1.4.6.4

[GUIDELINE] If a UPnP AV MediaServer exposes thumbnail images for image or video content, then a UPnP AV MediaServer may provide additional <res> elements for thumbnail images.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[46]	7V6GS	

7.4.1.4.6.5

[GUIDELINE] A UPnP A/V Media Server shall not expose a <res> element with a thumbnail media format profile ID (i.e. JPEG_TN, PNG_TN), without exposing at least one additional <res> element that is not one of the thumbnail media format profile IDs.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	O4VF4	

NOTE: Thumbnails are designed to augment other content items of any Media Class (Audio, AV, or Images) which can include non-DLNA contents. They are not meant to represent standalone images.

Images that will not be used as thumbnails but which match the thumbnail resolution will preferably be exposed using the appropriate smallest image media format profile ID (e.g. JPEG_SM).

7.4.1.4.7 MM DIDL-Lite AudioItem Album Art

7.4.1.4.7.1

[GUIDELINE] If a UPnP AV MediaServer exposes a CDS object with a <upnp:class> designation of object.item.audioItem or object.container.album.musicAlbum (or any class derived from either class), then the UPnP AV MediaServer should provide a <upnp:albumArtURI> element to present the URI for the album art.

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[46]	ZVDY7	

NOTE: Unlike image or video content, thumbnails for audio content will preferably be presented through the <upnp:albumArtURI> element.

7.4.1.4.7.2

[GUIDELINE] If a UPnP AV MediaServer exposes one or more <upnp:albumArtURI> elements for a single CDS object, then at least one of the URI values should point to thumbnail album art conforming to guideline 7.1.7 in [56].

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[46] [56]	YXRZ4	

NOTE: If album art thumbnails are provided, the desired expectation is to have JPEG thumbnails. Additional thumbnails can also be provided.

7.4.1.4.7.3

[GUIDELINE] If a UPnP AV MediaServer exposes a <upnp:albumArtURI> element with a URI pointing to a thumbnail conforming to a DLNA media format profile, then the <upnp:albumArtURI> shall have the albumArtURI@dlna:profileID attribute that identifies the DLNA profile ID of the thumbnail.

The namespace for DLNA defined properties shall be "urn:schemas-dlna-org:metadata-1-0/" and the namespace prefix shall be "dlna:".

EXAMPLE:

```
<upnp:albumArtURI dlna:profileID="JPEG_TN"
       xmlns:dlna="urn:schemas-dlna-org:metadata-1-0/">
   http:/192.168.1.1/album/albumArt1.jpg
</upnp:albumArtURI>
```

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	XRZ4Y	

NOTE: This guideline allows control points a convenient way to identify thumbnails that conform to a DLNA media format profile.

7.4.1.4.8 MM IFO File

7.4.1.4.8.1

[GUIDELINE] If a UPnP AV MediaServer exposes a content binary profiled according to MPEG-2 AV Format, Usage of Profile IDs, Profiles: MPEG_PS_NTSC and MPEG_PS_PAL, [56], (MPEG_PS_NTSC or MPEG_PS_PAL profiles), then the UPnP AV MediaServer shall either ensure that there are no SCR and/or PTS discontinuities (as defined in 7.4.1.4.8.8) or generate an IFO file if it detects discontinuity.

Note that this guideline does not apply in the scenario where a DMS exposes a content binary that was imported by the DMS, except when such content is received from a DLNA source.

All guidelines in 7.4.1.4.8 do not apply in scenarios involving URIs and/or <res> elements for RTP Media Transport because RTP encapsulation and padding make the offsets in the IFO file inaccurate.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [56]	VDY7V	

NOTE: Some decoders cannot handle the SCR/PTS discontinuous PS stream without proper additional decoder-specific control. This guideline provides the method that allows the DMP to obtain the information about the SCR/PTS discontinuous regions in program stream-profiled content.

The following are examples of content where a UPnP AV Media Server needs to comply with this guideline when exposing such content:

- A DMS application running on an open platform, such as a PC, that directly records, generates, or edits content.

The following are examples of content where a UPnP AV Media Server does not have to comply with this guideline when exposing such content:

- A DMS application running on an open platform, such as a PC, that receives or copies content from a non-DLNA source (e.g. Internet)
- A DMS application running on an open platform, such as a PC, which has other separate applications that record, create, or edit content or that import content from non-DLNA sources (e.g. Internet).

Although the above comments give examples where this guideline does not apply, it is still strongly encouraged that vendors provide IFO files in all cases when exposing an MPEG2 PS profiled content binary that has a discontinuity.

7.4.1.4.8.2

[GUIDELINE] If a UPnP AV MediaServer exposes a content binary profiled according to MPEG_PS_NTSC or MPEG_PS_PAL profiles ([56]), along with an associated IFO file, then it shall also expose the IFO file through the res@dlna:ifoFileURI attribute to present the URI for the IFO file as defined in 7.4.1.4.8.9.

The namespace for DLNA defined properties is "urn:schemas-dlna-org:metadata-1-0/" and the namespace prefix is "dlna:".

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[33] [46] [48] [56]	4VF4V	C

7.4.1.4.8.3

[GUIDELINE] If a Push Controller serves a content binary profiled according to MPEG_PS_NTSC or MPEG_PS_PAL profiles ([56]), along with an associated IFO file, directly to a UPnP AV MediaRenderer, then the Push Controller shall include the res@dlna:ifoFileURI attribute in the CurrentURIMetaData input argument for the AVT:SetAVTransportURI request as defined in 7.4.1.6.8.3. The res@dlna:ifoFileURI attribute contains the URI for the IFO file as defined in 7.4.1.4.8.9.

The namespace for DLNA defined properties is "urn:schemas-dlna-org:metadata-1-0/" and the namespace prefix is "dlna:".

[ATTRIBUTES]

M	A	+PU+	n/a	n/a	[33] [46] [48] [56]	LQ8HI	N

7.4.1.4.8.4

[GUIDELINE] If a UPnP AV MediaServer exposes a content binary profiled according to MPEG_PS_NTSC or MPEG_PS_PAL profiles ([56]), without SCR or PTS discontinuities, the UPnP AV MediaServer may provide a res@ dlna:ifoFileURI attribute to present the URI for the IFO file as defined in 7.4.1.4.8.9.

The namespace for DLNA defined properties is "urn:schemas-dlna-org:metadata-1-0/" and the namespace prefix is "dlna:".

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[33] [46] [48] [56]	V6GS6	C

NOTE: An IFO file can contain other metadata which might be useful to the Rendering Endpoint.

7.4.1.4.8.5

[GUIDELINE] If a Push Controller serves a content binary, profiled according to MPEG_PS_NTSC or MPEG_PS_PAL profiles ([56]), without SCR or PTS discontinuities, directly to a MediaRenderer, then the Push Controller may include the res@dlna:ifoFileURI attribute in the CurrentURIMetaData input argument for the AVT:SetAVTransportURI request as defined in 7.4.1.6.8.3. The res@dlna:ifoFileURI attribute contains the URI for the IFO file as defined in 7.4.1.4.8.9.

The namespace for DLNA defined properties is "urn:schemas-dlna-org:metadata-1-0/" and the namespace prefix is "dlna:".

[ATTRIBUTES]

O	A	+PU+	n/a	n/a	[33] [46] [48] [56]	PAHZ5	N

7.4.1.4.8.6

[GUIDELINE] If an IFO file is provided for a MPEG_PS_NTSC or MPEG_PS_PAL profiled content binary via an associated res@dlna:ifoFileURI attribute and/or the transport layer (such as described in 7.5.4.3.2.11), Rendering Endpoints shall render the content item even if the MPEG stream contains discontinuous SCR and/or PTS.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP M-DMD	n/a	n/a	23HQ6	

NOTE: If a MediaServer does not provide an IFO file (e.g. res@dlna:ifoFileURI is omitted) and a content binary, profiled according to MPEG-2 AV Format, Usage of Profile IDs, Profiles: MPEG_PS_NTSC and MPEG_PS_PAL, [56], has discontinuities, then the Rendering Endpoint can choose not to render the content binary.

7.4.1.4.8.7

[GUIDELINE] If a Rendering Endpoint attempts to render a content binary that is profiled according to MPEG-2 AV Format, Usage of Profile IDs, Profiles: MPEG_PS_NTSC and MPEG_PS_PAL, [56] that has discontinuities and no IFO file is available, then the Rendering Endpoint shall gracefully recover from the failure condition caused by any discontinuity.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP M-DMD	n/a	[56]	ZQRDX	

7.4.1.4.8.8

[GUIDELINE] An SCR or PTS discontinuity is defined as the occurrence of one of the following conditions in an MPEG2 PS content binary.

Condition1:

- If SCR(0) + SCRMaxValue - SCR(-1) <= 0.7 (to cover wrap-around case)
  SCR(0) + SCRMaxValue < SCR(-1) + PackDuration
- Otherwise SCR(0) < SCR(-1) + PackDuration

Or

Condition2:

- If PTS(-1) > PTS(0) (to cover wrap-around case)
  PTS(0) + PTSMaxValue - PTS(-1) > 0.61sec
- Otherwise PTS(0) - PTS(-1) > 0.61sec

Where

- SCR(0) is the SCR of the current pack.
- SCR(-1) is the SCR of the preceding pack.
- SCRMaxValue is 2^32 *300
- PackDuration is int((PackSize * 27000000) / (program_mux_rate * 50) = int((2048*27000000)/(25200 * 50)) = 43885
- PTS(0) is the PTS of the first picture of current GOP.
- PTS(-1) is the PTS of the first picture of preceding GOP.
- PTSMaxValue is 2^32

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[5] [6]	PXMLN	

NOTE: ISO/IEC13818-1 2.7.1, there is the following definition,

"The Program Stream shall be constructed such that the time interval between the bytes containing the last bit of system_clock_refrence_base fields in successive packs shall be less than or equal to 0,7s."

Note that SCR_base and PTS are limited to 32bit in the guideline.

Note that all the mathematical equations on calculating SCR/PTS discontinuity are expected to be performed using unsigned integer arithmetic as described in ISO/IEC 13818-1 subclause 2.4.3.7

7.4.1.4.8.9

[GUIDELINE] If a UPnP AV MediaServer provides an IFO file, then the URI of the IFO file shall be specified.

IFO file URIs are governed by guidelines in 7.4.1.3.10, except that the maximum length for an IFO file URI is 900 bytes.

Example:

- <res protocolInfo="http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC" duration="02:45:00" dlna:ifoFileURI="http://192.168.0.1:8080/IFO_101.ifo" xmlns:dlna="urn:schemas-dlna-org:metadata-1-0/">
  http://192.168.0.1:8080/MPEG/ntsc001.mpg
  </res>

In this case , the URI of the IFO file is http://192.168.0.1:8080/IFO_101.ifo

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	W9OGW

NOTE: Content Receivers can get the IFO file by issuing HTTP GET requests using this URI.

The reason for a shorter IFO file URI is because they are included in HTTP headers where the length is constrained.

7.4.1.4.9 MM CDS Browse/Search Action: Filter Argument

7.4.1.4.9.1

[GUIDELINE] The following five metadata properties shall always be present in the DIDL-Lite response, even if the metadata properties are not specified in the Filter argument of a CDS:Browse or CDS:Search request.

- @id
- @parentID
- @restricted
- dc:title
- upnp:class

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	AGPNV

NOTE: The Filter argument of the CDS:Browse and CDS:Search action instructs a UPnP AV MediaServer ContentDirectory to return only the specified metadata properties in the DIDL-Lite response of the *Result* output argument. This guideline clarifies that some metadata properties are required to be present even if they are not specified in the Filter argument.

7.4.1.4.9.2

[GUIDELINE] If an element of metadata property is specified, then the required attributes of the metadata element shall be presented.

For example:

- If a control point specifies "res" in the Filter, then res@protocolInfo is returned.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	9GG5U	

7.4.1.4.9.3

[GUIDELINE] A UPnP AV MediaServer control point should explicitly specify the desired metadata properties in the *Filter* input argument of a CDS:Browse or CDS:Search request.

[ATTRIBUTES]

S	R	DMP DMC +DN+ +UP+ +PR2+	M-DMP M-DMC M-DMU, M-DMD	MIU	[46] [49]	FH2V5	

NOTE: This guideline recommends control points to limit the requested metadata to only the metadata that will be used by the control point. A Filter value of asterisk "*" will likely cause the UPnP AV MediaServer to send more metadata than what the control point can actually use.

7.4.1.4.9.4

[GUIDELINE] In conjunction with 7.4.1.4.9.1, a UPnP AV MediaServer device shall not return metadata properties unless specified in the Filter argument.

For example:

- If a control point does not specify res@importUri in the Filter, then it is not returned.

Please note that having an attribute property in the Filter automatically requires the MediaServer to return the element to which the attribute belongs.

For example:

- If the control point specifies res@importUri (without "res"), then the "res" is also returned.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46] [49]	YK54Y	

7.4.1.4.9.5

[GUIDELINE] A UPnP MediaServer shall return DLNA metadata (i.e. attributes or elements with the dlna: prefix) only when the Filter argument indicates a request for the particular DLNA attribute(s) or element(s).

Note that "*" indicates a request for all attributes and elements.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[49]	K54YH	

NOTE: This behavior is required by the ContentDirectory specification and is consistent with the guideline 7.4.1.4.9.4.

Guidelines that require DLNA-defined metadata do not overrule underlying rules specified by the UPnP AV ContentDirectory service.

For example, guideline 7.4.1.4.7.3 requires albumArtURI@dlna:profileID, but a MediaServer only includes albumArtURI@dlna:profileID in a response when the Filter argument indicates a request for it.

7.4.1.4.9.6

[GUIDELINE] A UPnP MediaServer should declare the dlna: namespace (i.e. "urn:schemas-dlna-org:metadata-1-0/") only when the Filter argument indicates a request for one or more

DLNA attributes or elements and one or more DLNA attributes are included in the DIDL-Lite response.

Note that "*" indicates a request for all attributes and elements.

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[49]	H2V52	

NOTE: This guideline reduces the computational requirements of control points that employ validating schemas.

Although some guidelines require DLNA-defined elements and attributes in certain situations, the DLNA schema does not play a syntax enforcement role as all DLNA-defined elements and attributes are considered optional from the schema's perspective.

For example, guideline 7.4.1.4.7.3 albumArtURI@dlna:profileID when the associated URI points to an image that is compliant to a DLNA media format profile. Since it is impossible for the DLNA schema to determine if a URI points to a DLNA media format profile, the albumArtURI@dlna:profileID is considered optional attribute (from a schema perspective) even though it is required from a guidelines perspective.

7.4.1.4.10 MM CDS Browse/Search Action: Reduced Response Behavior

7.4.1.4.10.1

[GUIDELINE] A UPnP AV MediaServer device may reduce the number of CDS objects (<item> and <container> elements) in a response to a CDS:Browse or CDS:Search for the following scenarios only:

- The transmission of a SOAP response with a huge byte length (>204,800 bytes).
- The transmission of a SOAP response that exceeds 30 seconds for the transmission time.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[46] [49]	GG5U7	

NOTE: This guideline allows a UPnP AV MediaServer to limit the number of CDS objects returned in the SOAP response, even if the control point specified a desire for more CDS objects in the *RequestedCount* input argument. The reason for permitting such behavior is to allow UPnP AV MediaServer implementations to comply with other guidelines: 7.3.2.15 and 7.3.2.9.

7.4.1.4.10.2

[GUIDELINE] The number of CDS object entries (total <item> and <container> elements) in the *Result* output argument (containing the DIDL-Lite metadata) shall match the value specified in the *NumberReturned* output argument.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	GPNVY	

NOTE: This guideline will be followed, even if a UPnP AV MediaServer reduces the number of CDS objects returned in the SOAP response.

7.4.1.4.10.3

[GUIDELINE] If a UPnP AV MediaServer device reduces the number of CDS objects in a CDS:Browse(BrowseDirectChildren) or CDS:Search response then the number of returned CDS objects (as parsed in Result) shall be equal to the value of NumberReturned, which is less than RequestedCount.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	9OGWH	

NOTE: A UPnP AV MediaServer that limits the number of CDS objects is obligated to return a NumberReturned value that is consistent with the *RequestedCount* input argument.

7.4.1.4.10.4

[Guideline] A UPnP AV MediaServer control point shall use a RequestedCount of 0 or 1 and *StartingIndex* of 0 when using CDS:Browse with the *BrowseMetadata* option.

[Attributes]

M	L	DMP DMC +DN+ +UP+ +PR2+	M-DMP M-DMC M-DMU, M-DMD	MIU	[46] [49]	QRDX5	

NOTE: Improves expectations for CDS:Browse scenarios with *BrowseMetadata*.

7.4.1.4.10.5

[Guideline] A UPnP AV MediaServer device shall always return one CDS object (as indicated in TotalMatches and Result) when successfully responding to a CDS:Browse request with the *BrowseMetadata* option, regardless of the RequestedCount value.

[Attributes]

M	C	DMS	M-DMS	n/a	[46] [49]	3HQ6W	

7.4.1.4.10.6

[Guideline] If the UPnP AV MediaServer device returns more than zero CDS objects in a response to a CDS:Browse or CDS:Search query and if the UPnP AV MediaServer device does not provide an accurate value for the *TotalMatches* output argument, then the TotalMatches output value shall be set to zero.

[Attributes]

M	C	DMS	M-DMS	n/a	[46] [49]	6GS6Q	

NOTE: This guideline allows control points to conclude that a TotalMatches==0 condition indicates that the UPnP AV MediaServer could not accurately calculate the value in cases where the UPnP AV MediaServer actually returned CDS objects.

Although some UPnP AV MediaServer implementations might choose to report the accurate TotalMatches value, at the expense of violating the 30-second timeout rule, DLNA does not encourage that implementation option. The 7.3.2.9 guideline indicates that a control point is permitted to terminate a SOAP response that exceeds a 30-second transmission time.

7.4.1.4.10.7

[Guideline] If a UPnP AV MediaServer device cannot find more than zero CDS objects (in 27 seconds, as described in 7.3.2.9.2), for a response to a CDS:Browse or CDS:Search query and if the UPnP AV MediaServer cannot calculate an accurate value for the *TotalMatches* output argument, then the UPnP AV MediaServer should return a SOAP error response code of 720 (Cannot process the request).

[Attributes]

S	C	DMS	M-DMS	n/a	[46] [49]	VF4V2	

NOTE: This guideline covers the scenario where a UPnP AV MediaServer can neither find any CDS objects that satisfy the query nor calculate the *TotalMatches* output argument accurately. Although some UPnP AV MediaServer implementations might choose to report the accurate TotalMatches value, at the expense of violating the 27 seconds timeout rule, such behavior is not encouraged for the same reason stated in the previous guideline.

7.4.1.4.10.8

[GUIDELINE] A UPnP AV MediaServer control point should specify the desired number of CDS objects in the *RequestedCount* input argument of a CDS:Browse or CDS:Search query.

[ATTRIBUTES]

S	C	DMP DMC +DN+ +UP+ +PR2+	M-DMP M-DMC M-DMU, M-DMD	MIU	[46] [49]	DY7V3

NOTE: This guideline recommends control points to request a reasonable number of CDS objects in a single CDS query. The number of CDS objects that can be displayed to the user at a single time is a good measure of reasonableness. Using a RequestedCount of zero might cause the transmission of a huge SOAP response, which is undesirable.

7.4.1.4.10.9

[GUIDELINE] A UPnP AV MediaServer control point should specify smaller (about 10 to 30) RequestedCount input values for CDS:Browse and CDS:Search requests to receive a faster response time.

[ATTRIBUTES]

S	C	DMP DMC +DN+ +UP+ +PR2+	M-DMP M-DMC M-DMU, M-DMD	MIU	[46] [49]	RZ4YU

NOTE: Generally speaking, control points that specify smaller RequestedCount values will receive the response from the device sooner than if a larger value were specified.

7.4.1.4.10.10

[GUIDELINE] A UPnP AV MediaServer control point shall not assume that a UPnP MediaServer will return all of the CDS objects requested in a Browse request.

[ATTRIBUTES]

M	R	DMP DMC +DN+ +UP+ +PR2+	M-DMP M-DMC M-DMU, M-DMD	MIU	[46] [49]	Z4YU4

NOTE: This places requirements on an AV MediaServer control point to not assume that its Browse request will return all of the CDS objects it requested.

7.4.1.4.10.11

[GUIDELINE] If a UPnP AV MediaServer control point wants to retrieve the remaining items in a reduced response, the UPnP AV MediaServer control point shall issue additional Browse requests to complete the original Browse request for CDS objects.

[ATTRIBUTES]

M	R	DMP DMC +DN+ +UP+ +PR2+	M-DMP M-DMC M-DMU, M-DMD	MIU	[46] [49]	Y7V3Y

7.4.1.4.11 MM Container Update IDs Event

[GUIDELINE] UPnP AV MediaServer devices should implement behavior for the CDS.ContainerUpdateIDs state variable.

[ATTRIBUTES]

S	L	DMS	M-DMS	n/a	[46]	F4V2V

NOTE: This guideline is a benefit to both devices and control points, although lightweight UPnP AV MediaServer ContentDirectory Service (CDS) implementations might have difficulty implementing it. The rationale for this guideline stems from the fact that UPnP AV MediaServer control points can rely on the CDS.ContainerUpdateIDs state variable to minimize the number of CDS:Browse requests. A control point that relies solely on the CDS.SystemUpdateID state variable will browse the entire CDS hierarchy. Use of the CDS.ContainerUpdateIDs state variable can limit the browse requests to the container objects that observed the metadata changes.

7.4.1.4.12 MM Search Capabilities

7.4.1.4.12.1

[GUIDELINE] UPnP AV MediaServer entities that implement CDS:Search should support search queries with the following metadata properties:

- dc:title
- dc:creator
- upnp:class
- res@protocolInfo
- @refID

[ATTRIBUTES]

S	L	DMS	M-DMS	n/a	[46]	GS6QV

NOTE: This guideline describes the recommended search capabilities for a UPnP AV MediaServer that supports search operations.

7.4.1.4.12.2

[GUIDELINE] All searchable properties shall be listed in the return value of CDS:GetSearchCapabilities if the device implements CDS:Search.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	HQ6WO

NOTE: This guideline mandates that supported search properties be discoverable.

7.4.1.4.12.3

[GUIDELINE] If a UPnP AV MediaServer supports CDS:Search for an indicated property or a property with an indicated type, then it shall minimally support the following operators for the specified property types. Note that rows in the table are not additive.

Table 20 — CDS:Search Minimum Support of Operators

Property	Operators
@id	=, exists
@refID	= , exists
upnp:class	derivedfrom, exists
Any date, time, duration-based property types	< , <= , >= , > , = , !=, exists
All other string-based property types other than those listed in previous table entries	contains , =, exists
All URI-based property types	contains , =, exists
Integer or numerical property types	< , <= , >= , > , = , !=, exists
Boolean-based property types	= , !=, exists

Property	Operators
All other attributes and elements	exists

Vendors are free to apply additional CDS-normative operators for these properties or property types.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46]	RDX57	

NOTE: This guideline specifies the minimum behavior and capabilities for various query operators.

Vendors cannot change (override) the default behavior of these required operations as specified; but additions of standard operators are allowed.

Note that the exists operator only allows a value of "true" or "false" and those values are not quoted when used.

- VALID: @refID exists false
- INVALID: @refID exists 0
- INVALID: @refID exists "false"

7.4.1.4.12.4

[GUIDELINE] If a UPnP AV MediaServer reports a searchable property for a particular data type, then a UPnP AV MediaServer shall implement the associated operators for that data type.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46]	MLNQA	

NOTE: For example, if a UPnP AV MediaServer reports its search capabilities to be only dc:title, then it only needs to implement the exists, contains, and = operators.

7.4.1.4.13 MM Search All CDS Objects for a Media Class

7.4.1.4.13.1

[GUIDELINE] A UPnP AV MediaServer that implements the CDS:Search action should support the search with the following input parameters.

- ContainerID: 0
- *SearchCriteria*: upnp:class derivedfrom "[a upnp class for a supported media class]" and @refID exists false.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	OGWHP	

NOTE: A ContentDirectory service can expose CDS objects in different collections, such as, lists by genres, lists by artists, lists of favorite items, etc. In such cases, reference items, which have @refID are used to represent CDS items that are references to other CDS items in the CDS hierarchy.

In some use case scenarios, a UPnP AV MediaSerer control point wants to locate a set of CDS objects, excluding their duplicate references. For example, a UPnP AV MediaServer control point regenerates another presentation of CDS objects by using metadata properties of CDS objects. In this case, this search can be used.

Also note that that live AV and audio tuner content are also located by searching for object.item.videoItem.videoBroadcast and object.item.audioItem.audioBroadcast (and their derived classes). Note that the search results do not include the CDS container (i.e. ContainerID) that was specified in the request.

7.4.1.4.13.2

[GUIDELINE] If the UPnP AV MediaServer supports a DLNA media class, it shall support a corresponding object class as follows:.

Table 21 — UPnP:class for searching all CDS objects

Media Class	upnp:class value
Audio	object.item.audioItem
Image.	object.item.imageItem
AV	object.item.videoItem

The following is an example to search all video items which are stored in the ContentDirectory service.

- request: Search("0", "upnp:class derivedfrom "object.item.videoItem" and @refID exists false", "dc:date,upnp:genre,res@duration", 0, 40, "")
- response:

```
Search("<DIDL-Lite xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/" xmlns="urn:schemas-upnp-
org:metadata-1-0/DIDL-Lite/">
<item id="10" parentID="4" restricted="0">
<dc:title>Desire stones</dc:title>
<dc:date >2004-07-04T20:00:00</dc:date>
<upnp:genre>Movie</upnp:genre>

<upnp:class>object.item.videoItem</upnp:class>
<res protocolInfo="http-get:*:
video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC;DLNA.ORG_FLAGS=01100000000000000
000000000000000" duration="0:59:53">http://192.168.1.1/res?id=10
</res>
    </item>
        <item id="13" parentID="12" restricted="0">
        <dc:title>Music Street in Asia</dc:title>
        <dc:date>2004-07-04T23:30:00</dc:date>
        <upnp:genre>Music</upnp:genre>

<upnp:class>object.item.videoItem</upnp:class>
<res protocolInfo="http-get:*:
video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC;DLNA.ORG_FLAGS=01100000000000000
000000000000000" duration="0:29:54">http://192.168.1.1/res?id=13</res>
    </item>
</DIDL-Lite>",
2, 2, 10434)
```

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	n/a	PNVYT	

7.4.1.4.13.3

[GUIDELINE] UPnP AV MediaServer responses for CDS:Search should only list CDS containers once. Duplicate CDS containers should be omitted from the results set

Duplicate CDS containers are CDS containers that provide the same set of child CDS items.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	G5U7Z	

NOTE: Duplicate CDS containers can be implemented in a ContentDirectory service which exposes the same content in different groupings.

For example, a music library can expose the same set of music tracks in two different CDS containers. In this example, the CDS containers could have a hypothetical path from the root as follows.

- "0" => "All Albums" => "The Doors Greatest Hits"
- "0" => "All Artists" => "The Doors" => "The Doors Greatest Hits"

This guideline recommends the MediaServer to return either of these containers rather than both of them in the CDS:Search response where the SearchCriteria is "upnp:class derivedfrom "object.container.musicAlbum"".

7.4.1.4.13.4

[GUIDELINE] The UPnP AV MediaServer that supports the recommended search criteria of guideline 7.4.1.4.13.1 shall return a SearchCaps output value (from CDS:GetSearchCapabilities) that includes upnp:class and @refID.

Example:

- request: GetSearchCapabilities
- response: GetSearchCapabilities("upnp:class,@refID")

The ContentDirectory service of this MediaServer shall provide the following properties and the values as metadata of the root Container.

- @searchable="1"
- <upnp:searchClass includeDerrived="1">[upnp class]</upnp:class> or <upnp:searchClass includeDerrived="0">[upnp class]</upnp:class>

The following DIDL-Lite fragment is an example of metadata for a root Container in the ContentDirectory service that supports Audio, Image and AV media classes.

Example:

-
```
<DIDL-Lite xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/" xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/">
<container id="0" parentID="-1" childCount="3" restricted="1" searchable="1">
    <dc:title>Root Container</dc:title>
    <upnp:class>object.container</upnp:class>
    <upnp:searchClass
        includeDerived="1">object.item.audioItem</upnp:searchClass>
    <upnp:searchClass
        includeDerived="1">object.item.imageItem</upnp:searchClass>
    <upnp:searchClass
        includeDerived="1">object.item.videoItem</upnp:searchClass>
</container>
</DIDL-Lite>
```

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	2V52H

NOTE: A UPnP AV MediaServer control point can know whether a ContentDirectory service supports the search defined in guideline 7.4.1.4.13.1 by checking the search capabilities and the metadata of the root container.

7.4.1.4.14 MM Keyword Search Templates

[GUIDELINE] A UPnP AV MediaServer that implements the CDS:Search action should support the following types of CDS:Search requests, with the *ContainerID* input parameter set to "0".

- "dc:title contains " val1 " and @refID exists false"
- "dc:creator contains " val1 " and @refID exists false"
- "upnp:album contains " val1 " and @refID exists false"

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	54YHU	

NOTE: Some control points with an alphanumeric input mechanism (keyboards, virtual keyboards, cell phone keypad, etc.) can benefit from keyword searching capabilities.

7.4.1.4.15 MM/BT Basic Tuner Container

7.4.1.4.15.1

[GENERAL] DLNA defines two tuner implementations. They are indicated as either a Basic Tuner or an Extended Tuner. The Basic Tuner guidelines, as defined in 7.4.1.4.15 - 7.4.1.4.22 by the initial publishing of the Design Guidelines, are based upon the UPnP AV MediaServer:1 specifications and those guidelines have been relabeled as “MM/BT Basic Tuner”. The Extended Tuner guidelines, as defined in 7.4.5, are based upon the TUNER Feature defined in UPnP AV MediaServer:2 and higher specifications

7.4.1.4.15.2

[GUIDELINE] If a UPnP AV MediaServer implements a Basic Tuner, then it shall conform to all of the requirements for a Basic Tuner as defined in 7.4.1.4.15.3 - 7.4.1.4.15.6 and 7.4.1.4.16 - 7.4.1.4.22.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[49]	4BI9R	N

NOTE: It is optional for a UPnP AV MediaServer to implement a Basic Tuner. The Basic Tuner implementation can be implemented by any version of the UPnP AV specifications.

7.4.1.4.15.3

[GUIDELINE] A UPnP AV MediaServer tuner should be represented as a container with a class of object.container or any derived class. A Basic Tuner container should have an associated name. The name is given by property dc:title.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	4YHUX	E

NOTE: It allows multiple tuners to be represented in CDS with a friendly name. Tuner channel ordering allows the control point to implement up/down by selecting the next item in the container.

See Annex B for recommendations on how to represent a turner container.

Control points: note that in compliance with UPnP Guidelines (7.4.1.4.9), conformant DMS devices will not return the dlna:containerType property unless it is specifically requested as part of a CDS:Browse Filter argument.

7.4.1.4.15.4

[GUIDELINE] A UPnP AV MediaServer Basic Tuner container shall have a dlna:containerType property and the property shall have a value of Tuner_1_0.

The name space for DLNA defined properties shall be "urn:schemas-dlna-org:metadata-1-0/" and the namespace prefix shall be "dlna:".

The following is an example.

- <dlna:containerType xmlns:dlna="urn:schemas-dlna-org:metadata-1-0/">Tuner_1_0</dlna:containerType>

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	V52H7	E

7.4.1.4.15.5

[GUIDELINE] A UPnP AV MediaServer Basic Tuner container shall contain object items of class object.item.videoItem.videoBroadcast or object.item.audioItem.audioBroadcast or both. (Objects derived from either class also qualify.)

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	5U7ZK	E

7.4.1.4.15.6

[GUIDELINE] The order of the object items (order of <item> elements in a CDS:Browse response)in a UPnP AV MediaServer Basic Tuner container should correspond to the tuner up/down operation. For example, channel number or channel preset order.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	NVYTV	E

7.4.1.4.16 MM/BT Basic Tuner Audio Tuner

[GUIDELINE] If a UPnP AV MediaServer provides live audio content from a tuner, the UPnP AV MediaServer should use object.item.audioItem.audioBroadcast or any derived class as the upnp:class value.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	GWHPZ	

NOTE: These guidelines allow control points to identify content sourced from a tuner.

7.4.1.4.17 MM/BT Basic Tuner Video Tuner

[GUIDELINE] If a UPnP AV MediaServer provides live video or audio/video content from a tuner, the UPnP AV MediaServer should use object.item.videoItem.videoBroadcast or any derived class as the upnp:class value.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	LNQA9	

NOTE: See comment text for 7.4.1.4.16.

7.4.1.4.18 MM/BT Basic Tuner Properties: Channel Title

[GUIDELINE] A UPnP AV MediaServer tuner dc:title property should describe the program content if available otherwise should contain the contents of the upnp:channelName CDS property. If upnp:channelName CDS property is not available the upnp:channelNr property should be used.

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[46]	DX57C	E

NOTE: This is to clarify the meaning of title in context of a tuner. Some vendors might interpret title as channel name.

7.4.1.4.19 MM/BT Basic Tuner Properties: Channel Number

7.4.1.4.19.1

[GUIDELINE] A UPnP AV MediaServer broadcast object item should have the associated property upnp:channelNr.

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[46]	Q6WOI	E

NOTE: This guideline clarifies the intended usage of upnp:channelNr.

7.4.1.4.19.2

[GUIDELINE] If a UPnP AV MediaServer CDS object contains the upnp:channelNr property, then each upnp:channelNr number shall be unique within the context of its container.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46]	S6QVG	E

NOTE: For usages where the same upnp:channelNr number indicates different media formats for the same content (e.g.. SD and HD content), this would be realized by using the recommended practice in DLNA to use multiple <res> elements in the same CDS object 7.4.1.4.3.

7.4.1.4.20 MM/BT Basic Tuner Properties: Channel Name

[GUIDELINE] If a UPnP AV MediaServer broadcast object item has the associated property upnp:channelName, then each upnp:channelName string should be unique within the context of its container. The upnp:channelName should be used to identify the channels, not the program content.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	4V2VO	E

NOTE: This guideline clarifies the intended usage of upnp:channelName.

7.4.1.4.21 MM/BT Basic Tuner Content URI

[GUIDELINE] The channel selection and the connection to the tuner are invoked through the connection establishment to the URI of the resource associated with the broadcast object item.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	7V3YS	

NOTE: This guideline essentially requires tuner content to be advertised and accessible through a URI.

7.4.1.4.22 MM/BT Basic Tuner Control Point Assumptions

[GUIDELINE] The UPnP AV MediaServer control point should not assume that the currently viewed channel is the channel that it previously selected. Due to possible sharing of the tuner by multiple clients the channel can change without the client being aware of the change.

[ATTRIBUTES]

S	A	DMP DMC +DN+	M-DMP M-DMC M-DMD	MIU	[46]	4YU4Z	

NOTE: Because there is no feed back mechanism between the tuner and the control point it is not possible to know the current channel when there are multiple Content Receivers connected as clients or if the tuner is being used by a local output device.

7.4.1.4.23 MM TakeOut Contents

7.4.1.4.23.1

[GUIDELINE] A UPnP AV MediaServer may provide the DLNA defined property<dlna:takeOut> for a CDS object that belongs to a group of objects intended for unidirectional synchronization (from the server to a downloading endpoint).

The <dlna:takeOut > is a non-empty string value with a maximum length of 256 bytes.

The name space for DLNA defined properties shall be "urn:schemas-dlna-org:metadata-1-0/" and the namespace prefix shall be "dlna:".

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46]	YU4ZM	

NOTE: The guidelines for takeOut enable unidirectional synchronization scenarios. This allows a device to easily find a group of content that is intended for download. Control points find groups by using CDS:Search and specifying the <dlna:takeOut> property with the desired group name.

The general assumption is that a control point (bundled with a Download Controller or an M-DMD) will find content with a particular group name and download it (unless it has already acquired it). The <dlna:takeOut> tag will generally persist on the DMS or M-DMS, such that if a CDS object is no longer part of a group, then the control point will know to remove the content from its local storage during the synchronization operation.

The process of assigning the value of the <dlna:takeOut> tag is vendor defined. One way is for the DMS or M-DMS to assign the value for a user when the user creates the group. Another way is to allow the user to name the group. An extremely simple way is to assign the same group name to all CDS objects.

DLNA suggests that DMS or M-DMS implementations provide a way for a user to specify a maximum storage quota when creating a group. This type of user interface feature is useful because users are generally aware that downloading endpoints have storage limitations for downloading groups.

Note that this guideline does not imply any requirement that the downloader mirrors the CDS hierarchy when it downloads content from the CDS. This is especially true when the takeOut group includes CDS containers (i.e. the CDS container has the <dlna:takeOut> element.

7.4.1.4.23.2

[GUIDELINE] A CDS object (advertised by a UPnP AV MediaServer) may contain more than one <dlna:takeOut> element.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46]	V3YS5	

NOTE: A CDS object can belong to multiple groups.

This guideline provides for basic unidirectional synchronization of content from a DMS to UPnP Control Points. The primary mechanism used by a control point to identify content between reboots or application launches is through the object ID. The mechanism to indicate if content has been modified is to change the object ID but to change the value of the URI (i.e. value of the <res> element).

7.4.1.4.23.3

[GUIDELINE] If a UPnP AV MediaServer provides the <dlna:takeOut> property to a CDS object and wants to indicate that the content is identical between reboots or application launch/shutdown, then it shall not change the object ID of the CDS object.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	V2VOM	

7.4.1.4.23.4

[Guideline] If a UPnP AV MediaServer control point is synchronizing a particular take-out group, then under normal test conditions the control point should only transfer content binaries belonging to CDS objects that are part of the take-out group.

(i.e. If a UPnP AV MediaServer control point is not able to find a take-out CDS object that was synchronized on a previous session or if the CDS object of the same @id value no longer has the <dlna:takeOut> property with the same group name, then the control point should assume that the CDS object is no longer part of the take-out group.

[Attributes]

S	A	+DN+	M-DMD	MIU	[46]	6QVGY	

NOTE: The phase under normal test conditions means no resource and network limitations.

7.4.1.4.23.5

[Guideline] If a UPnP AV MediaServer control point is synchronizing a particular take-out group and finds a take-out CDS object that was not synchronized on a previous session, then under normal test conditions it should attempt to transfer at least one content binary from the associated CDS object.

(i.e. If a UPnP AV MediaServer control point finds a take-out CDS object that was not synchronized on a previous session, then it should assume that the CDS object is a new member of the take-out group.)

[Attributes]

S	A	+DN+	M-DMD	MIU	[46]	6WOIW	

NOTE: The phase under normal test conditions means no resource and network limitations.

7.4.1.4.23.6

[Guideline] If a UPnP AV MediaServer provides the <dlna:takeOut> property to a CDS object and wants to indicate that a content binary is modified (e.g. editing), it shall change the @id value to a new and unique value, relative to the entire CDS hierarchy.

[Attributes]

M	C	DMS	M-DMS	n/a	[46]	X57CY	

7.4.1.4.23.7

[Guideline] If a UPnP AV MediaServer supports the <dlna:takeOut> property, then it shall support the DLNA- defined CDS:X_GetTakeOutGroupNames action.

The action's definition in the service description is:

```
<action>
   <name>X_GetTakeOutGroupNames</name>
   <argumentList>
      <argument>
         <name>GroupNames</name>
         <direction>out</direction>
         <relatedStateVariable>
            X_A_ARG_Type_GroupNames
         </relatedStateVariable>
      </argument>
   </argumentList>
```

```
</action>
```

The X_A_ARG_TYPE_GroupNames state variable is defined:

```
<stateVariable sendEvents="no">
   <name>X_A_ARG_Type_GroupNames</name>
   <dataType>string</dataType>
</stateVariable>
```

The *GroupNames* output argument is a comma-separated value list of all take-out group names.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	NQA9W	

NOTE: The CDS:GetTakeOutGroupNames action allows a control point to easily acquire the list of group names for use in a CDS:Search request. DLNA has two normative search templates for finding groups. The only difference between the two is that vendors can specify a upnp:class value if desired.

7.4.1.4.23.8

[GUIDELINE] If a UPnP AV MediaServer supports the <dlna:takeOut> property, then it shall support searching the CDS as defined in guideline 7.4.1.4.13 and the additional searches with the following input parameters.

- Additional Search #1 input parameters::
  - ContainerID = 0
  - *SearchCriteria*: @refID exists false and dlna:takeOut="[group name]"
- Additional Search #2 input parameters::
- ContainerID = 0
- *SearchCriteria*: upnp:class derivedfrom "[a upnp class for a supported media class]" and @refID exists false and dlna:takeOut="[group name]"

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	WHPZV	

7.4.1.4.23.9

[GUIDELINE] A UPnP AV MediaServer that supports the search defined in guideline 7.4.1.4.23.8 shall return the SearchCaps that includes the <dlna:takeOut> element to the CDS:GetSearchCapablities action.

Example:

- request: CDS:GetSearchCapabilities()
- response: CDS:GetSearchCapabilities("upnp:class,@refID, dlna:takeOut")

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	VYTV3	

NOTE: A UPnP AV MediaServer control point can know whether a ContentDirectory service supports the search defined in guideline 7.4.1.4.23.8 by checking the search capabilities.

7.4.1.4.24 MM CDS Containers and Media Collection Binaries

7.4.1.4.24.1

[GUIDELINE] Whenever possible a UPnP AV MediaServer should use a container object with a set of child item objects to indicate a media collection.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	U7ZKE	

NOTE: Using this model for media collections allows rendering devices to render media collections. Media collection file formats are useful in various ways, but v1.0 DMP's might lack the ability to parse media collection files.

7.4.1.4.24.2

[GUIDELINE] A UPnP AV MediaServer may associate a <res> element with a CDS container or item such that the <res> element's URI value points to a media collection file conforming to a media format profile defined in Media Collection Profile Guidelines clause 11 of [56] (e.g. DIDL_S, DIDL_V).

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46] [56]	52H79	

NOTE: Media collection files are a convenient way to allow a rendering device (e.g. DMR) to play a media collection.

7.4.1.4.24.3

[GUIDELINE] A UPnP AV MediaServer that advertises a <res> element for a media collection file should also use the res@dlna:trackTotal attribute, which is a ui4 value.

This guideline applies only to media collection files that are normative to the DLNA guidelines (e.g. DIDL_S and DIDL_V).

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	YHUXA	

NOTE: Controlling devices (e.g. DMC, M-DMC) can use this value when the calculating track index for the currently playing content in a DLNA PlayContainer URI operation.

7.4.1.4.24.4

[GUIDELINE] If present, the value of res@dlna:trackTotal shall equal the number of content entries (a.k.a sequenced tracks) in the media collection file.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	HUXAV	

7.4.1.4.24.5

[GUIDELINE] If a UPnP AV MediaServer wants to convey that the items (or tracks) in a media collection file have a 1:1 mapping to the child CDS objects in a particular container, then the UPnP AV MediaServer should associate the <res> element (for the media collection file) with the CDS container that has the child CDS objects. The ordering of the 1:1 mapping is the same for both the media collection file and the CDS objects in the CDS container (as obtained from a CDS:Browse request with an unspecified sorting criteria).

This methodology should be used even when the media collection file uses or references content intended for presentation-layer details. A UPnP AV MediaServer that uses this methodology claims an intent that the content in the media collection file maps to the child objects of the associated container.

Note that CDS container in this guideline means any CDS object with the object.container or similarly derived upnp:class designation.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	2H79M

NOTE: This guideline urges UPnP AV MediaServer that wants to represent the child CDS objects encapsulated by a CDS container to associate a playlist file with the parent CDS container.

This guideline is only a recommendation because the file format of the media collection file could reference things intended for the presentation layer. For example, a slideshow file might have URI values that point to proprietary background music instructions and transitions. In such a case, the MediaServer associates the slideshow file with a container that was the parent of image items because the general intent is to have a 1:1 mapping.

7.4.1.4.24.6

[GUIDELINE] If a UPnP AV MediaServer wants to convey that the items (or tracks) in media collection file do not have a direct mapping to a particular container, then the MediaServer should associated the <res> element (for the media collection file) with a CDS item that has the object.item.playlistItem (or similarly derived upnp:class) designation.

This methodology should be used whenever the media collection file has an intentional mapping of content in multiple CDS containers. Likewise, this methodology should be used when the media collection file references content (that is not associated with presentation-layer details).

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	7ZKEU

NOTE: In cases where a vendor wants to provide a playlist file without implying any intent about the mapping of that content to a particular CDS container, then the vendor is encouraged to associate the media collection file with a CDS item object. This particular model (while easier for the DMS or M-DMS) can disadvantage rendering devices (e.g. DMP) that do not support media collection files because they rely solely on the CDS hierarchy to provide the rendering experience.

7.4.1.4.25 MM Rendering Media Collection Files

7.4.1.4.25.1

[GUIDELINE] A Rendering Endpoint may support rendering of media collection files.

[ATTRIBUTES]

O	A	DMP DMR	M-DMP M-DMD	n/a	[48]	YTV3Y

NOTE: DMP, M-DMP, M-DMD, and DMR devices are not required to render media collection files.

7.4.1.4.25.2

[GUIDELINE] A UPnP AV MediaRenderer should support playback of a DLNA PlayContainer URI.

[ATTRIBUTES]

S	A	DMR	n/a	n/a	[48]	HPZV4

NOTE: DMR devices are not required to support rendering of a DLNA PlayContainer URI because the feature requires the DMR to have a MediaServer control point. However, this feature is recommended for a DMR because

DLNA recommends that DMS devices expose media collections through CDS container objects that have a set of child CDS item objects.

Implementing this feature requires adherence to additional guidelines, such as those in 7.4.1.6.10 and 7.4.1.4.28.

7.4.1.4.25.3

[Guideline] If a UPnP AV MediaRenderer supports the DLNA PlayContainer URI operation, then it shall specify the <dlna:X_DLNACAP> element (as a child of the <device> element that represents the MediaRenderer) with the *playcontainer-depth-strict* token.

More formally, the syntax of the capability ID is defined below.

Table 22 — Capability ID Syntax

Capability ID	Description
playcontainer-depth-strict	The UPnP AV MediaRenderer supports the DLNA PlayContainer URI operation.

- "playcontainer-capability-id = "playcontainer" "-" depth-token "-" strict-token
- "depth-token = <ui4 value>
- "strict-token = "0" \| "1"

The "playcontainer" portion of the capability ID is a literal, string value.

The depth-token shall be a ui4 value, indicating the maximum depth the MediaRenderer will traverse when rendering a PlayContainer URI operation. A value of "0" means that no sibling containers of first-item-id-arg will be traversed.

The strict-token portion of the capability shall be a "0" or "1". If the value is "1" then MediaRenderer supports a URI syntax where first-item-id-arg shall be a an immediate child of container-id. If the value is "0" then MediaRenderer supports a URI syntax where first-item-id-arg shall be a descendent of container-id-arg and the parent of first-item-id-arg shall have a container depth that is less than or equal to depth.

M	A	DMR	n/a	n/a	[48]	QA9W8

NOTE: Guideline 7.3.2.35.1 provides the syntax for the <dlna:X_DLNACAP> element.

In addition to indicating that the DLNA PlayContainer URI operation is supported by the MediaRenderer, the capability ID also indicates the URI limitations that are imposed by the MediaRenderer. The depth portion of the syntax allows the MediaRenderer to restrict the CDS container depth of a media collection. The strict portion of the syntax allows the MediaRenderer to require the first played CDS item to be an immediate child of the media collection's top CDS containerSee 7.4.1.4.28 for more information on the DLNA PlayContainer URI syntax.

7.4.1.4.26 MM CDS DLNA PlaySingle URI Values

7.4.1.4.26.1

[Guideline] A UPnP AV MediaServer may have CDS item objects that have a <res> element with a DLNA PlaySingle URI.

[Attributes]

O	A	DMS	M-DMS	n/a	[46]	57CYQ

NOTE: A DLNA PlaySingle URI is a URI that allows a DMS or M-DMS to reference a CDS object on the same or different DMS or M-DMS. This can be used for a variety of things, including a media collection composed of content from multiple DMS or M-DMS devices or for DMS virtualization.

7.4.1.4.26.2

[Guideline] A <res> element with a DLNA PlaySingle URI value shall have a similar res@protocolInfo value as one of the <res> elements of the referenced CDS object.

A similar res@protocolInfo value has the following characteristics.

- The first field shall be a string in the form of "playsingle-"<transport>, where the transport shall be the DLNA transport identifier of the referenced <res> element.
- The 2nd, 3rd, and 4th fields shall be copied identically from the referenced <res> element.

Examples:

- playsingle-http-get:*:audio/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC; DLNA.ORG_OP=01;DLNA.ORG_FLAGS=01100000000000000000000000000000
- playsingle-rtsp-rtp-udp:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC; DLNA.ORG_OP=10;DLNA.ORG_PS=-1,2/3,4; DLNA.ORG_FLAGS=03100000000000000000000000000000;DLNA.ORG_MAXSP=9.75

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	WOIWZ

NOTE: DLNA PlaySingle URI values can only have a single protocolInfo value. Therefore, a single CDS object can have more than one <res> elements with the same DLNA PlaySingle URI value and the different protocolInfo values. This allows UPnP AV MediaServer control points to know the media formats that are available on the referenced CDS object.

7.4.1.4.26.3

[Guideline] A UPnP AV MediaServer may have more than one <res> element with the same DLNA PlaySingle URI associated with a CDS item object.

[Attributes]

O	A	DMS	M-DMS	n/a	[46]	QVGY4

7.4.1.4.26.4

[Guideline] A CDS item that has one or more DLNA PlaySingle URI <res> elements may have other <res> elements that do not have a DLNA PlaySingle URI as a value.

[Attributes]

O	A	DMS	M-DMS	n/a	[46]	2VOMJ

NOTE: A MediaServer's CDS item can mix <res> elements that have different types of URI values. For example, a virtual DMS might have DLNA PlaySingle URI values for original content served by a different DMS while using RTP and HTTP URI values for the converted content that the virtualizing DMS will serve.

7.4.1.4.26.5

[Guideline] A UPnP AV MediaServer shall only associate DLNA PlaySingle URI <res> elements with CDS item objects. DLNA PlaySingle URI <res> elements shall not be associated with CDS container objects. DLNA PlaySingle URI <res> elements shall not reference CDS container objects.

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	3YS5B

NOTE: DLNA PlaySingle URI values are allowed only when used in conjunction with audio, audio/video, or image content. Using DLNA PlaySingle URI <res> elements to reference CDS containers, media collection files, or other CDS objects with a DLNA PlaySingle URI <res> element can result in circular references.

7.4.1.4.26.6

[Guideline] The res@protocolInfo value of a DLNA PlaySingle URI <res> element shall reference content that qualifies as audio, audio/video, or image content.

A res@protocolInfo value that identifies a media collection file or XHTML Print document is prohibited. Similarly, a <res> DLNA PlaySingle URI value that references a media collection file is also prohibited.

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	U4ZMV	

7.4.1.4.26.7

[Guideline] A DLNA PlaySingle URI shall only reference a CDS item object that does not have a DLNA PlaySingle URI <res> element.

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	4ZMVW	

7.4.1.4.26.8

[Guideline] A DLNA PlaySingle URI shall only reference a CDS item object with a upnp:class that is the same or derived from one of the following:

- object.item.audioItem
- object.item.videoItem
- object.item.imageItem

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	YS5BS	

7.4.1.4.26.9

[Guideline] A CDS item that has a DLNA PlaySingle URI <res> element shall have the same upnp:class value as the CDS object that is referenced or one of its super classes.

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	VOMJU	

7.4.1.4.26.10

[Guideline] A DLNA PlaySingle URI may be used with media format profiles that are not defined by DLNA.

[Attributes]

O	A	DMS	M-DMS	n/a	[46]	VGY43	

NOTE: The DLNA PlaySingle URI values can be used with undefined media formats. In such scenarios, the DLNA PlaySingle URI value still need to comply with all of the rules declared in guideline 7.4.1.4.26.

7.4.1.4.26.11

[Guideline] The syntax of DLNA PlaySingle URI is as follows.

- dlna-playsingle-uri = "dlna-playsingle://" cds-udn "?" service-id-arg item-id-arg
- cds-udn = <the UDN of the UPnP AV MediaServer device>
- service-id-arg = "sid=" service-id-val
- service-id-val = <service ID of the CDS belonging to the MediaServer>

- item-id-arg = "&iid=" item-id-val
- item-id-val = <The @id string value of the CDS item to be referenced. The CDS item shall comply with guidelines 7.4.1.4.26.6 - 7.4.1.4.26.8>

PlaySingleURI values shall be case-sensitive values, even though the general URI syntax is a case-insensitive value. The URI shall also be escaped as described by 7.4.1.3.10.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	OIWZO

7.4.1.4.27 MM Control Point Rules for DLNA PlaySingle URIs

7.4.1.4.27.1

[GUIDELINE] A UPnP AV MediaRenderer control point shall not invoke AVT:SetAVTransportURI with the *CurrentURI* input argument set to a DLNA PlaySingle URI.

[ATTRIBUTES]

M	A	DMC +PU+	M-DMC	MIU	[44] [48]	7CYQB

NOTE: See 7.4.1.4.27.3 for more information.

7.4.1.4.27.2

[GUIDELINE] DLNA device classes and capabilities with a UPnP AV MediaServer control point and appropriate transport layer components may support the rendering or transporting of content referenced by DLNA PlaySingle URI values.

[ATTRIBUTES]

O	A	DMP DMR +DN+	M-DMP M-DMD	MIU	[49]	A9W8G

NOTE: DMP, DMR, and Download Controllers are not required to render or transport DLNA PlaySingle URIs.

7.4.1.4.27.3

[GUIDELINE] A UPnP AV MSCP+MRCP that wants to instruct a UPnP AV MediaRenderer to play a DLNA PlaySingle URI shall resolve the DLNA PlaySingle URI to a URI for the actual audio, audio/video, or image content for use with the AVT:SetAVTransportURI request.

[ATTRIBUTES]

M	A	DMC	M-DMC	MIU	[44] [48] [49]	PZV4Y

NOTE: Prior to invoking a AVT:SetAVTransportURI, the control point will set up a UPnP AV connection between the DMR and the Content Source. Since a DLNA PlaySingle URI is a URI pointer to a CDS object (and is not a pointer to an actual content binary), the MediaRenderer control point does not have sufficient information to create the logical connection prior to invoking AVT:SetAVTransportURI.

7.4.1.4.28 MM DLNA PlayContainer URI

7.4.1.4.28.1

[GUIDELINE] The syntax of a DLNA PlayContainer URI shall be as described in the BNF notation. Assume no linear white space.

- dlna-playcontainer-uri = "dlna-playcontainer://" cds-udn "?" service-id-arg container-id-arg first-item-id-arg first-item-index-arg [sort-arg] [max-depth-arg]
- cds-udn = <the UDN of the UPnP AV MediaServer device>
- service-id-arg = "sid=" service-id-val

- service-id-val = <service ID of the CDS belonging to the MediaServer>
- container-id-arg = "&cid=" container-id-val
- container-id-val = <The @id string value of the CDS container that represents the top container of the media collection.>
- first-item-id-arg = "&fid=" first-item-id-val
- first-item-id-val = <The @id string value of the first CDS item that will be played as part of the PlayContainer URI operation. Note that 7.4.1.4.25.3 (MM Rendering Media Collection Files) places restrictions on first-item-id-val. Specifically, if the strict-token (defined in 7.4.1.4.25.3) is "1", then first-item-id-val shall be a CDS item that is an immediate child of container-id-val. If strict-token is "0", then first item-id-val shall be a CDS item that is descended from container-id-val, whose parent container has a container depth that is less than or equal to the depth-token (defined in 7.4.1.4.25.3).>
- first-item-index-arg = "&fii=" first-item-index-val first-item-index-val = <a ui4 value, as used in a CDS:Browse request. This value shall represent the zero-based index of the first-item-id-val CDS item, such that a CDS:Browse request where the following argument values will result in a response where the first returned CDS item has @id = first=item-id-val.
    - *ObjectID* = parent container of first-item-id-val
    - *BrowseFlag* = "BrowseDirectChildren",
    - *SortCriteria* = sort-arg, and
    - *StartingIndex* = first-item-index-val
- sort-arg = "&sc=" sort-val
- sort-val = <A *SortCriteria* string, as defined for a CDS:Browse request. Equivalently, the value shall comply with the A_ARG_TYPE_SortCriteria syntax, defined by the ContentDirectory service. If sort-arg is omitted, then assume an effective value of an empty string.>
- max-depth-arg = "&md=" max-depth-val
- max-depth-val = <A ui4 value that specifies the maximum descent level in the container. If max-depth-arg is omitted, then the value infers an effective value of "0". Note that 7.4.1.4.25.3 (MM Rendering Media Collection Files) places restrictions on max-depth-arg. Specifically the max-depth-val shall be less than or equal to the depth-token, as defined in 7.4.1.4.25.3.>

The cds-udn, service-id-val, object-id-val, sort-val, object-index-val, and max-depth-val tokens shall be URI-escaped according to [29].

Ordering of tokens is significant.

The maximum length of a DLNA PlayContainer URI is 1024 bytes.

[ATTRIBUTES]

M	A	DMC +PU+	M-DMC	MIU	[29] [46] [49]	TV3Y4

NOTE: This guideline defines the syntax of a DLNA PlayContainer URI.

7.4.1.4.28.2

[GUIDELINE] If a UPnP AV MediaRenderer supports the DLNA PlayContainer URI operation, then the MediaRenderer shall support the sort-arg and max-depth-arg portions of the dlna-playcontainer-uri syntax.

- sort-arg: This value determines the playback order, such that the playback order matches the order as observed from an equivalent CDS:Browse response.

- max-depth-arg: This value determines the maximum depth the MediaRenderer traverses when encountering child/descendent containers. A value of "0" means that child/descendent are not traversed.

[Attributes]

M	C	DMR	n/a	n/a	[46] [49]	ZKEU4	

NOTE: These guidelines clarify that control points are not required to specify all parameters in a DLNA PlayContainer URI, but a DMR will operate correctly if a control point does provide the optional parameters.

Note that a DLNA PlayContainer URI does not necessarily mean that the DMR will issue a single type of CDS:Browse request. For example, the DMR's control point might progressively issue multiple CDS:Browse requests that result in fewer results to ensure that the 200 KB response limit is not exceeded by the DMS.

7.4.1.4.28.3

[Guideline] If a UPnP AV MediaRenderer supports the DLNA PlayContainer URI operation and the URI specifies a max-depth-val that is greater than depth-token, as defined in 7.4.1.4.25.3, then the MediaRenderer shall return a UPnP AV error code of 716 (Resource not found) in the AVT:SetAVTransportURI response.

[Attributes]

M	A	DMR	n/a	n/a	[49]	H79M4	

7.4.1.4.28.4

[Guideline] If performing a DLNA PlayContainer URI operation, a UPnP AV MediaRenderer shall only play CDS items as part of the DLNA PlayContainer URI and shall only render <res> elements of images, Audio-only, and AV content. This guideline works in conjunction with 7.4.1.3.31.1 (MM playcontainer-param (DLNA PlayContainer Flag), which prohibits the playcontainer-param from having a value of true for <res> elements of media collection binaries.

[Attributes]

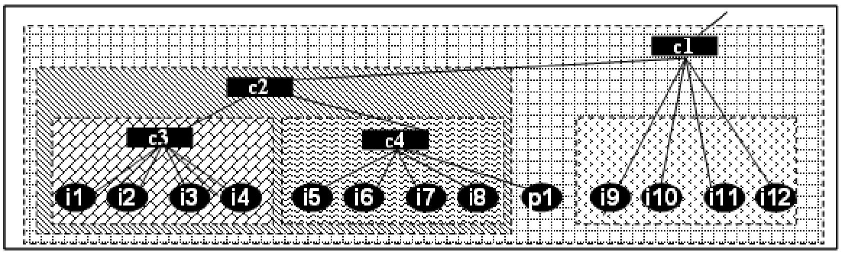
M	L	DMR	n/a	n/a	[48] [49]	UXAV4	

7.4.1.4.28.5

[Guideline] MediaRenderer control points may omit the sort-arg, and max-depth-arg tokens in the dlna-playcontainer-uri syntax.

[Attributes]

O	C	DMC +PU+	M-DMC	MIU	[48] [49]	XAV44	



Collection, start at	container-id-arg	first-item-id-arg	first-item-index-arg	max-depth-arg	Default Ordered tracks in collection (*e)	Track Index: item → AVT.CurrentTrack Index
"i1"	"c3"	"i1"	"0"	Value >= "0" or omitted	i1, i2, i3, i4	i1→1, i2→2, i4→4
"i6"	"c4"	"i6"	"1"	Value >= "0" or omitted	i6, i7, i8, i5 (*a)	i6→2, i8→4, i5→1
"c3"	This is not permitted because first-item-id-arg must be a CDS item. If first-item-id-arg was "i1", then playcontainer-strict must be false (See next line/playlist).					
"i1"	"c2"	"i1"	"0"	Value >= "1"	i1, i2, ... i4, i5, ... i8; (*a)	"i1"→1, "i8"→8
"i12"	"c1"	"i12"	"4"; (*b)	Value = "0" or omitted	i12, i9, i10, i11	"i12"→4, "i9"→1, "i11"→3
"i1"	"c1"	"i1"; (*c)	"0"	Value >= "2"	i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, i11, i12	"i1"→1, "i4"→14, "i5"→5, "i8"→8, "i9"→10, "i12"→13; (*d)
"i9"	"c1"	"i9"	"1"	Value >= "2"	i9, i10, i11, i12, i1, i2, i3, i4, i5, i6, i7, i8,	"i9"→10, "i12"→13, "i1"→1, "i4"→4, "i5"→5, "i8"→8; (*d)

(*a): Although "p1" is child of "c4", it is a CDS object with DIDL playlist file so it is automatically skipped.
(*b): "c2" is a child of "c1" that appears before "i9" when browsing the DMS.
(*c): this scenario is only permitted when playcontainer-strict is false.
(*d): tracks after "i9" have an index number that is off by one because of p1, which is a CDS object with a DIDL-Lite playlist file. Also note that CDS containers are not rendered during a PlayContainer URI operation.
(*e): DMR must follow the current AVT.PlayMode (i.e. in case of NORMAL play mode, it must stop after playing the track of maximum AVT.CurrentTrack).

Figure 28 — <DLNA PlayContainer URI Example>

7.4.1.4.29 MM Control Point Rules for DLNA PlayContainer URI

7.4.1.4.29.1

[GUIDELINE] A UPnP AV MediaRenderer Control Point may invoke AVT:SetAVTransportURI with the *CurrentURI* input argument set to a DLNA PlayContainer URI.

[ATTRIBUTES]

O	A	DMC +PU+	M-DMC	MIU	[44] [48]	79M4Q

NOTE: Notice that the usage of the DLNA PlayContainer URI requires the existence of a CDS at the content source. For this reason, a +PU+ can send the DLNA PlayContainer URI when its hosting Device Class contains a CDS or can interact with a CDS on the network.

7.4.1.4.29.2

[GUIDELINE] If a UPnP AV MediaRenderer Control Point that invokes AVT:SetAVTransportURI with the *CurrentURI* input argument set to a DLNA PlayContainer URI, then the UPnP AV MediaRenderer Control Point shall only do so with UPnP AV MediaRenderer devices that have at least one protocolInfo value in the SinkProtocolInfo that has the playcontainer-param flag set to true.

SinkProtocolInfo refers specifically to the *Sink* output argument of CMS:GetProtocolInfo responses and the CMS.SinkProtocolInfo state variable.

[ATTRIBUTES]

M	A	DMC +PU+	M-DMC	MIU	[44] [45] [48]	KEU4V

NOTE: See 7.4.1.3.23.2 and 7.4.1.3.31 for more information.

7.4.1.5 Basic Connection Management (BCM) Guidelines

7.4.1.5.1 MM/BCM UPnP AV Connection Rules

7.4.1.5.1.1

[GUIDELINE] UPnP AV MediaServers may provide support for basic connection management (BCM).

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[45] [49]	V3Y4Y

NOTE: DMS devices are not required to support basic connection management (BCM). A DMS with BCM reports the existence of transport layer connections by using UPnP AV connections, which might be enumerated and terminated. This feature helps 3rd party controllers and DMPs offer the users the ability to manage the connections according to their own desires (mainly for termination). This feature does not require any control point to support Basic Connection Management in order to operate basic browse/play operations. This feature does not require a control point to interact with the user.

Transport clients are encouraged to implement these guidelines to allow MediaServers with BCM support to properly deliver BCM functionality.

- 7.5.4.3.2.36 MT/BCM HTTP Header:peerManager.dlna.org
- 7.5.4.3.2.37 MT/BCM HTTP Header:friendlyName.dlna.org
- 7.5.4.4.6.2.27 MT RTP/BCM RTSP peerManager.dlna.org
- 7.5.4.4.6.2.28 MT RTP/BCM RTSP friendlyName.dlna.org

The DLNA guidelines use the AVT:Stop action to terminate streams on a DMR instead of using CMS:ConnectionComplete.

7.4.1.5.1.2

[GUIDELINE] If a UPnP AV MediaServer supports the BCM, then it shall create a UPnP AV Connection with a unique ConnectionID for each new Transport Layer request corresponding to a content request, including Background, Interactive, and Streaming requests. (Creating UPnP AV Connections for upload-related content transfer process is out-of-scope.)

The list of Transport Layer requests corresponding to a media stream request includes HTTP GET requests and the RTSP SETUP command. The list of Transport Layer requests excludes HTTP HEAD and HTTP POST requests and all HTTP requests not related to a content URI (i.e. URI for a <res> element exposed by a UPnP AV MediaServer).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[45] [49]	ZV4YE

NOTE: The creation of UPnP AV Connection is a result of a Transport Layer (HTTP or RTSP) connection request for a new media stream. (See 7.4.1.5.2 for request for an existing media stream). All other non media stream connection requests do not require the creation of UPnP AV Connection. The media stream is identified by the content URI in the transport layer request.

CMS:PrepareForConnection is not required for the creation of a UPnP AV Connection. However, using CMS:PrepareForConnection in conjunction with the creation of a transport layer is a permissible way to create UPnP AV Connection. Note that a UPnP AV MediaServer control point is not required to call the optional CMS:PrepareForConnection. Hence a DMS with BCM that also implements CMS:PrepareForConnection will also be able to create the UPnP AV Connection without requiring a control point to call CMS:PrepareForConnection execution.

The UPnP AV Media Server preferably stores internally all necessary information along with the newly created ConnectionID for later actions (CMS:ConnectionComplete and CMS:GetCurrentConnectionInfo). For example, relevant informations are, but not limited to, protocolInfo, URI, and TCP/IP socket handles. The protocolInfo is already available in the DMS to satisfy 7.5.4.3.2.10.

7.4.1.5.1.3

[Guideline] A UPnP AV MediaServer that assigns a ConnectionID value in the range from "1" to "2147483647" should assign the value in an increasing manner with a rollover value of "1" when the vendor-defined-maximum-value is exceeded.

The vendor-defined-maximum-value is a vendor-defined value that is greater than or equal to "65535" and less than or equal to "2147483647".

[Attributes]

S	A	DMS	M-DMS	n/a	[45] [49]	9W8GK	

7.4.1.5.1.4

[Guideline] A UPnP AV MediaServer that assigns a ConnectionID value in the range from "1" to "2147483647" should start value assignment with a random value.

[Attributes]

S	A	DMS	M-DMS	n/a	[45] [49]	CYQBA	

7.4.1.5.1.5

[Guideline] If a UPnP AV MediaServer supports BCM, then it shall expose the UPnP AV Connections using CMS:GetCurrentConnectionIDs.

[Attributes]

M	A	DMS	M-DMS	n/a	[45] [49]	IWZOH	

7.4.1.5.1.6

[Guideline] If a UPnP AV MediaServer supports BCM, then it shall remove the ConnectionID from the list of connections provided in CMS:GetCurrentConnectionIDs, when all transport layer connections (associated with the ConnectionID) are closed/terminated.

[Attributes]

M	A	DMS	M-DMS	n/a	[45] [49]	GY43Y	

NOTE: The guideline clarifies the life-cycle of a ConnectionID. An active and advertised ConnectionID will be associated with at lease one active media transport connection.

7.4.1.5.1.7

[GUIDELINE] If a UPnP AV MediaServer supports BCM, then it shall maintain the CMS.CurrentConnectionIDs state variable and deliver an event for the ConnectionManager service to signal that an update to the connection information has occurred.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[45] [49]	OMJU4

NOTE: The guideline repeats the UPnP AV requirements.

7.4.1.5.1.8

[GUIDELINE] If a UPnP AV MediaServer supports BCM, then it shall implement CMS:GetCurrentConnectionInfo for all UPnP AV Connections reported by CMS:GetCurrentConnectionIDs.

The following arguments and values are returned by CMS:GetCurrentConnectionInfo:

- RemoteProtocolInto contains the res@protocolInfo of the transported content on the given connection. The value corresponds to the protocolInfo used to set the contentFeatures.dlna.org HTTP header or RSTP header according to 7.5.4.3.2.10 and 7.5.4.4.6.2.66.
- PeerConnectionManager contains
  - an empty string if the information is not available, or
  - the PeerConnectionManager value provided by a DMR using the peerManager.dlna.org HTTP header (defined in 7.5.4.3.2.36) or the peerManager.dlna.org RTSP header (defined in 7.5.4.4.6.2.27).
  - the friendly name value provided by a DMP or M-DMP (that implements this version or a newer version of the DLNA interoperability guidelines) using the friendlyName.dlna.org HTTP header (defined in 7.5.4.3.2.37) or the friendlyName.dlna.org RTSP header (defined in 7.5.4.4.6.2.28).
- PeerConnectionID is set to "-1".
- Direction is set to "OUT".

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[45] [49]	S5BS2

NOTE: This guideline specifies the values returned by CMS:GetCurrentConnectionInfo for a given ConnectionID. Most arguments are compliant with the UPnP AV Architecture with the exeption of PeerConnectionManager, which is not available froM-DMP implementations that adheres to v1.0 of the DLNA Interoperability Guidelines.

CMS:PrepareForConnection can also provide the PeerConnectionManager value when executed previously for the specifed UPnP AV Connection.

7.4.1.5.1.9

[GUIDELINE] If a UPnP AV MediaServer supports BCM, then it shall use the following notation to advertise a DMP or M-DMP user friendly name (when provided) in the PeerConnectionManager parameter, returned by CMS:GetCurrentConnectionInfo.

- peerconnectionmanager_arg = "fnam" ":" friendly-name "/"
- friendly-name = <string, limited to 128 bytes in its UTF-8 encoded form>

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[45] [49]	ZMVW7

NOTE: The DMP user friendly name is provided on the Media Transport layer connection (see 7.5.4.3.2.37 or 7.5.4.4.6.2.28) and is used to identify the Content Receiver connected to the DMS or M-DMS.

The format of the PeerConnectionManager is different from the UPnP Architecture in such a way that it cannot be confused with a non-existing CMS Service (i.e: no CMS on DMP).

Example: "fnam:LivingRoom TV/" where "fnam" replaces "uuid" and the ServiceId token is left empty.

7.4.1.5.1.10

[GUIDELINE] If a UPnP AV MediaServer supports BCM, then it shall implement CMS:ConnectionComplete.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[45] [49]	MVW74	

NOTE: The requirement for CMS:ConnectionComplete does not imply to support and implement CMS:PrepareForConnection in the ConnectionManager service.

CMS:ConnectionComplete is the UPnP action that instructs a UPnP MediaServer to tear down a UPnP AV Connection (and its underlying transport layer connection) when it is no longer required. When CMS:ConnectionComplete is executed for a given ConnectionID, the device is expected to respond with a success and terminate the associated HTTP or RTSP transport connections related to the UPnP AV Connection. Subsequently, all resources related to the connection are released. This behavior is similar to a termination initiated by the connection creator on the Content Receiver endpoint. When a connection is terminated (e.g. DMP performs a Stop media operation, AVT:Stop is invoked on a DMR, the transport layer connection is broken, CMS:ConnectionComplete is called, etc.) it is expected that proper resource clean up takes place.

7.4.1.5.2 MM/BCM UPnP AV Connection Rules for HTTP

[GUIDELINE] If a UPnP AV MediaServer supports BCM, then it shall not create a new UPnP AV Connection when the ConnectionID is provided in the HTTP Request using the scid.dlna.org header. Instead it shall reuse the provided ConnectionID.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	n/a	5BS2H	

NOTE: This guideline provides a mechanism to identify multiple transport layer request separated in time but related to the same media stream (identified by the URI). For example, during a pause operation and/or consecutive random access requests the HTTP Client might indicate in the HTTP Get request that all transport layer requests are related to the same media stream which does not require a new ConnectionID. With such mechanism the UPnP AV MediaServer is not required to maintain a local table containing most recent requested URI with the associated ConnectionID per HTTP client. It is the HTTP client's responsibility to indicate the relationship between HTTP requests.

7.4.1.5.3 MM/BCM UPnP AV Connection Rules for RTP

[GUIDELINE] If a UPnP AV MediaServer supports BCM and it supports RTP Media Transport, then it shall not create a new UPnP AV Connection when the ConnectionID is provided in the RTSP session-id header. Instead it shall reuse the provided ConnectionID.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	n/a	MJU4Z	

7.4.1.5.4 MM/BCM-DMS CMS:ConnectionComplete and Closing Transport Connections

7.4.1.5.4.1

[GUIDELINE] If a UPnP AV MediaServer supports BCM, then when it responds successfully to a CMS:ConnectionComplete request for a valid ConnectionID it shall close all related Transport Layer connections.

Likewise, the corresponding ConnectionID shall be removed from the list of connections provided in CMS:GetCurrentConnectionIDs.

[Attributes]

M	A	DMS	M-DMS	n/a	n/a	Y43YT	

NOTE: This guideline clarifies the expected behavior when UPnP MediaServers response successfully to a CMS:ConnectioComplete. The guideline does not prohibit a UPnP MediaServer to return an error code (Eg. 704 Local Restricyion) to indicate the impossibility to close the Transport Layer connections.

7.4.1.5.4.2

[Guideline] If a UPnP AV MediaServer supports BCM, then it shall return 704 Local Restriction to the CMS:ConnectionComplete request if it does not accept to close all related Transport Layer connections.

[Attributes]

M	A	DMS	M-DMS	n/a	n/a	WZOH4	

7.4.1.6 MediaRenderer Device Requirements

7.4.1.6.1 MM UPnP AV MediaRenderer CMS:GetProtocolInfo Behavior

7.4.1.6.1.1

[Guideline] A UPnP AV MediaRenderer that lists a protocolInfo in its SinkProtocolInfo shall be able to render that type of content when given a URI for that type of content.

SinkProtocolInfo refers specifically to the *Sink* output argument of CMS:GetProtocolInfo responses and the CMS.SinkProtocolInfo state variable.

[Attributes]

M	C	DMR	n/a	n/a	[45] [48]	YQBA6	

NOTE: DMR devices that claim support for a DLNA media format profile will be able to render that content when instructed to do so by a DMC. A DMR device that supports playback of content but requires a control point to send a URI to a media collection file or a DLNA PlayContainer URI in the AVT:SetAVTransportURI request is not permissible.

7.4.1.6.1.2

[Guideline] A UPnP AV MediaRenderer that supports playback of media collection files shall include the media collection file's protocolInfo value in its list of SinkProtocolInfo (as observed from the *Sink* output argument in CMS:GetProtocolInfo responses and its CMS.SinkProtocolInfo state variable).

[Attributes]

M	A	DMR	n/a	n/a	[45] [48]	W8GKW	

NOTE: DMR uses protocolInfo values to report support for media collection files. Media collection files have mime types and media format profile IDs. DMR devices are not required to support playback of media collection files because the feature requires a DMR to implement a MediaServer control point.

7.4.1.6.2 MM AVTransport Default Instance

7.4.1.6.2.1

[Guideline] A UPnP AV MediaRenderer shall always have an AVTransport virtual instance identified with an InstanceID equal to "0".

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[44]	3Y4Y3	

NOTE: This represents the "default" instance of the AVTransport service. This allows UPnP AV MediaServer control points to assume that the virtual instance with InstanceID value of "0" is always available. There is no need to invoke CMS:PreparForConnection when using default instances.

7.4.1.6.2.2

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						EU4VW	

7.4.1.6.3 MM RenderingControl Default Instance

7.4.1.6.3.1

[GUIDELINE] A UPnP AV MediaRenderer shall always have a RenderingControl virtual instance identified with an InstanceID equal to "0".

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[51]	9M4Q9	

NOTE: This represents the "default" instance of the RenderingControl service. This allows UPnP AV MediaRenderer control points to assume that the virtual instance with InstanceID value of "0" is always available. There is no need to invoke CMS:PrepareForConnection when using default instances.

7.4.1.6.3.2

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						AV44K	

7.4.1.6.4 MM AVTransport Multiple Instances

[GUIDELINE] A UPnP AV MediaRenderer may have multiple virtual instances of the AVTransport service.

[ATTRIBUTES]

O	R	DMR	n/a	n/a	[45]	V44KU	

NOTE: Multiple virtual instances of the AVTransport service will include the default instance (InstanceID=0) as per guideline 7.4.1.6.2.1. Multiple virtual instances of the AVTransport service are permitted but DLNA defines no interoperability guidelines for InstanceIDs that are not equal to "0".

7.4.1.6.5 MM RenderingControl Multiple Instances

[GUIDELINE] A UPnP AV MediaRenderer may have multiple virtual instances of the RenderingControl service.

[ATTRIBUTES]

O	R	DMR	n/a	n/a	[45]	M4Q9J	

NOTE: Multiple virtual instances of the RenderingControl service will include the default instance (ID=0) as per guideline 7.4.1.6.3.1. Multiple virtual instances of the RenderingControl service are permitted but DLNA defines no interoperability guidelines for InstanceIDs that are not equal to '0'.

7.4.1.6.6 MM LastChange

7.4.1.6.6.1

[GUIDELINE] UPnP MediaRenderers shall implement the AVT.LastChange and RCS.LastChange evented state variables as specified in [44] and [51].

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[44] [51]	U4VW4

NOTE: All AVT and RCS state variables, except AVT.RelativeTimePosition, AVT.AbsoluteTimePosition, AVT.RelativeCounterPosition and AVT.AbsoluteCounterPosition, are evented indirectly through the AVT.LastChange and RCS.LastChange state variables as described in [51].

7.4.1.6.6.2

[GUIDELINE] If any of the instance state variables in the AVTransport Service or Rendering Control Service, except AVT.RelativeTimePosition, AVT.AbsoluteTimePosition, AVT.RelativeCounterPosition and AVT.AbsoluteCounterPosition, have changed, a UPnP MediaRenderer shall send an event containing the corresponding LastChange state variable. The content of the LastChange state variable shall contain the latest value of the state variables updated since the last event. The event shall be sent immediately, or 0.2 seconds after the last event was sent, whichever is later.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [51]	95C3Y

NOTE: This guideline enables control points to rely on UPnP events to track the playback status of the MediaRenderer device.

The Max Event Rate of the AVT.LastChange and RCS.LastChange state variables is one event every 0.2 seconds, which translates to a maximum of five events per second, or a minimum of 0.2 seconds between successive events.

As an example, if the values of AVT.TransportState and AVT.TransportURI instance state variables of instance 0 of the AVTransport service change to "TRANSITIONING" and "http://192.168.0.1:8080/MPEG/ntsc001.mpg" respectively, the value of the AVT.LastChange state variable in an event sent by the AVTransport service contains the following text (prior to XML-escaping).

```
<Event xmlns="urn:schemas-upnp-org:metadata-1-0/AVT/">
  <InstanceID val="0">
    <TransportState val="TRANSITIONING" />
    <AVTransportURI val="http://192.168.0.1:8080/MPEG/ntsc001.mpg" />
  </InstanceID>
</Event>
```

Similarly, if the values of RCS.Brightness and RCS.Contrast instance state variables of instance 0 of the Rendering Control service change to "20" and "50" respectively, the value of the RCS.LastChange state variable in an event sent by the Rendering Control service contains the following text (prior to XML-escaping).

```
<Event xmlns="urn:schemas-upnp-org:metadata-1-0/RCS/">
  <InstanceID val="0">
    <Brightness val="20 " />
    <Contrast val="50" />
  </InstanceID>
</Event>
```

7.4.1.6.6.3

[GUIDELINE] UPnP MediaRenderers shall not send AVT or RCS events with empty values for the LastChange state variable, except when sending the first event after a subscription.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44] [51]	QR39T	

NOTE: Sending empty events is a waste of both network and device resources.

7.4.1.6.7 MM LastChange Frequency

[GUIDELINE] UPnP MediaRenderers should send no more than one AVT.LastChange or RCS.LastChange event for a single AVTransport or RenderingControl service action.

[ATTRIBUTES]

S	A	DMR	n/a	n/a	[44] [51]	Y4Y37	

NOTE: Although UPnP AV has a maximum frequency for sending AVTransport and RenderingControl events, the preference is to send fewer events to reduce the load on control points.

7.4.1.6.8 MM AVT:SetAVTransportURI

7.4.1.6.8.1

[GUIDELINE] If a UPnP AV MediaRenderer cannot accept an AVT:SetAVTransportURI request because the current value of the AVT.TransportState virtual instance state variable is not "STOPPED" or "NO_MEDIA_PRESENT", then it shall return an error 705 (Transport is Locked).

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44] [48]	4YEOC	

NOTE: This guideline requirement is both an addition and limiting guideline to [44].

If a UPnP AV MediaRenderer control point encounters the 705 error code, then it can invoke AVT:Stop to stop the transport layer and then retry the AVT:SetAVTransportURI request.

There can be reasons for the UPnP AV MediaRenderer to return an error, even when the transport state is STOPPED.

7.4.1.6.8.2

[GUIDELINE] A UPnP AV MediaRenderer control point shall provide an empty string or a valid URI value, as defined in 7.4.1.3.10, in the *CurrentURI* input argument for an AVT:SetAVTransportURI request.

[ATTRIBUTES]

M	C	DMC +PU+	M-DMC	n/a	[44] [48]	QQ8QS	

NOTE: This clarifies that an empty string can be used as a URI value in the *CurrentURI* input argument to clear the rendering state.

7.4.1.6.8.3

[GUIDELINE] If the *CurrentURI* input argument for the AVT:SetAVTransportURI request is not an empty string or a DLNA PlayContainer URI, then the *CurrentURIMetaData* input argument of the same request shall be a valid value, as defined in 7.4.1.6.14.9.

[ATTRIBUTES]

M	A	DMC +PU+	M-DMC	n/a	[44] [48]	USRWQ	

NOTE: This is a mandate over what UPnP normally allows as optional behavior.

For a Push Controller (+PU+) that does not contain a CDS, the DIDL-Lite metadata can typically be created from a DIDL-Lite XML fragment template containing only the minimal properties as described in 7.4.1.6.14.9. For example, since the @id property value only needs to be unique within the scope of the DIDL-Lite XML fragment, the @id property value can be any value chosen by the Push Controller; the @parent property can have a value of -1, and the @restricted property value can be either 0 or 1.

7.4.1.6.8.4

[Guideline] If the *CurrentURI* input argument for the AVT:SetAVTransportURI request contains the URI of a media collection file, then the *CurrentURIMetaData* input argument of the same request shall contain only one <res> element and its URI value shall be the same as the value contained in the *CurrentURI* input argument.

[Attributes]

M	A	DMC +PU+	M-DMC	n/a	[44] [48]	G3HWW

NOTE: Metadata for a media collection is only meaningful for a single <res> CDS object.

7.4.1.6.8.5

[Guideline] If the *CurrentURI* input argument for the AVT:SetAVTransportURI request contains a DLNA PlayContainer URI, then the *CurrentURIMetaData* input argument of the same request shall be an empty string.

[Attributes]

M	A	DMC +PU+	M-DMC	n/a	[44] [48]	OVM6G

NOTE: Metadata has no meaning for the DLNA PlayContainer URI case.

7.4.1.6.8.6

[Guideline] If the *CurrentURI* input argument for the AVT:SetAVTransportURI request is an empty string, then the *CurrentURIMetaData* input argument of the same request should be an empty string.

[Attributes]

S	A	DMC +PU+	M-DMC	n/a	[44] [48]	3HWWT

NOTE: Metadata has no meaning when the *CurrentURI* input argument is an empty string (i.e. no content to be rendered). The behavior for the UPnP AV MediaRenderer is vendor dependent when metadata is provided (some UPnP AV MediaRenderers will select a <res> element from the metadata to render as per guideline requirement 7.4.1.6.9.1; other UPnP AV MediaRenderers will reset the rendering state.)

7.4.1.6.9 MM UPnP AV MediaRenderer Selects a Different <res> Element

7.4.1.6.9.1

[Guideline] A UPnP AV MediaRenderer that receives an AVT:SetAVTransportURI request may override the *CurrentURI* input argument by selecting one of the <res> elements specified in the *CurrentURIMetaData* input argument.

[Attributes]

O	A	DMR	n/a	n/a	[44] [48]	8GKWS

NOTE: The general purpose is to allow a UPnP AV MediaRenderer control point to recommend a URI using the *CurrentURI* input argument while providing alternate choices to the UPnP AV MediaRenderer through the *CurrentURIMetaData* input argument of the AVT:SetAVTransportURI action.

If a UPnP AV MediaServer exposes multiple <res> elements, the UPnP AV MediaRenderer control point might not be able to determine the preferred <res> element for a given server/renderer pair. Examples of multiple <res> elements include:

- HTTP and RTP Media Transports

- Transcoded content
- Multi-homed content
- PS and ES versions of content (RTP Media Transport only)
- Transrated content available at different bitrates

The CMS:GetProtocolInfo action alone does not provide sufficient information for a UPnP AV MediaRenderer control point to always select the most appropriate URI to pass to AVT:AVTransportSetURI.

7.4.1.6.9.2

[Guideline] If a UPnP AV MediaRenderer control point calls AVT:SetAVTransportURI and provides a DIDL-Lite XML fragment for the *CurrentURIMetaData* argument, then the DIDL-Lite fragment should include all of the <res> elements associated with the object represented by the DIDL-Lite XML fragment.

[Attributes]

S	L	DMC +PU+	M-DMC	MIU	[44] [48]	ZOH44

NOTE: The DIDL-Lite XML fragment will at least have the <res> element for the URI specified in request, as required by 7.4.1.6.14.9. Additional <res> elements are encouraged to allow the UPnP AV MediaRenderer to choose a different URI for playback. In addition the DIDL-Lite XML fragment when sourced from a UPnP AV MediaServer is from a CDS object.

7.4.1.6.10 MM UPnP AV MediaRenderer & DLNA PlayContainer URI

7.4.1.6.10.1

[Guideline] A UPnP AV MediaRenderer that supports playback of a DLNA PlayContainer URI shall use a playcontainer-param set to true in the 4th field protocolInfo parameter of individual protocolInfo to indicate that the UPnP AV MediaRenderer will render that type of content when given a DLNA PlayContainer URI. This guideline specifically applies to the protocolInfo values listed in the *Sink* output argument of CMS:GetProtocolInfo responses and the CMS.SinkProtocolInfo state variable.

All guideline requirements in 7.4.1.6.10 apply in the context of a UPnP AV MediaRenderer that supports the rendering of DLNA PlayContainer URIs.

All guideline requirements in 7.4.1.6.10 apply in conjunction with all other guideline requirements that govern a UPnP AV MediaServer control point.

[Attributes]

M	A	DMR	n/a	n/a	[45] [48]	43YT3

NOTE: A UPnP AV MediaRenderer indicates support for DLNA PlayContainer URI values by using the playcontainer-param flag. ProtocolInfo values that omit this parameter indicate that the UPnP AV MediaRenderer will not play that type of content as part of a DLNA PlayContainer URI operation.

Setting the playcontainer-param flag to true does not mean that a UPnP AV MediaRenderer requires the media type to be played only through a DLNA PlayContainer URI operation. (i.e. UPnP AV MediaRenderer control point can specify a URI that points to content of the specified protocolInfo or the UPnP AV MediaRenderer control point can specify a DLNA PlayContainer URI that will result in the playback of content with the specified protocolInfo.)

7.4.1.6.10.2

[Guideline] If a UPnP AV MediaRenderer supports playback of a DLNA PlayContainer URI, then it shall support playback of a DLNA PlaySingle URI as part of the DLNA PlayContainer URI operation.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[48]	JU4ZA	

NOTE: A UPnP AV MediaRenderer that can play a DLNA PlayContainer URI already has a UPnP AV MediaServer control point that enables it to browse a UPnP AV MediaServer.

7.4.1.6.10.3

[GUIDELINE] A UPnP AV MediaRenderer that supports rendering of DLNA PlayContainer URIs shall support rendering of a CDS object, if the CDS object meets the following criteria.

- The CDS object falls into the media collection that is defined by the DLNA PlayContainer URI.
- The CDS object has a <res> element with a res@protocolInfo that indicates a supported media format profile, as described by 7.4.1.6.1.2.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44] [48]	BS2HV	

NOTE: This guideline essentially says that a UPnP AV MediaRenderer will render content that is supported by the UPnP AV MediaRenderer and can be found by traversing the CDS in the manner prescribed for a DLNA PlayContainer URI.

7.4.1.6.11 MM UPnP AV MediaRenderer AVT State Variables

7.4.1.6.11.1

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						VW74M	

7.4.1.6.11.2

[GUIDELINE] .A UPnP AV MediaRenderer that renders a content binary that is not part of a DLNA PlayContainer URI or media collection file shall use the following mapping for virtual instance state variables.

- The AVT.AVTransportURI value shall be the URI provided in the *CurrentURI* input argument of the AVT:SetAVTransportURI request or it shall be a URI obtained from a <res> element in the *CurrentURIMetaData* input argument of the same request.
- The AVT.CurrentTrackURI value shall be equal to the AVT.AVTransportURI value.
- The AVT.CurrentTrack and AVT.NumberOfTracks values shall be "1".

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [48]	W74MV	

NOTE: In conjunction with other DLNA guidelines that define general rules for ensuring consistency between different AVTransport state variables, this guideline clarifies how those other guidelines apply specifically, in the case where a control point instructs a UPnP AV MediaRenderer to render a single content binary.

Guideline requirement 7.4.1.6.9.1 permits a UPnP AV MediaRenderer to override the *CurrentURI* input argument with a URI from the *CurrentURIMetaData* input argument.

7.4.1.6.11.3

[GUIDELINE] A UPnP AV MediaRenderer that renders a media collection file shall use the following mapping for virtual instance state variables.

- The AVT.AVTransportURI value shall be the URI provided in the *CurrentURI* input argument of the AVT:SetAVTransportURI request.
- The AVT.CurrentTrackURI value shall be the URI for the track that is currently being rendered. (i.e. the URI for the actual audio-only, audio/video, or image content that is actually rendered)
- The AVT.CurrentTrack value shall be the index within the media collection file, where the URI for the track can be found. This value is 1 for the first track URI in the media collection file and is incremented by one for each successive track even if the track cannot be rendered.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [48]	S2HVW

NOTE: In conjunction with other DLNA guidelines that define general rules for ensuring consistency between different AVTransport state variables, this guideline clarifies how those other guidelines apply specifically, in the case of media collection files. There could be some tracks in the media collection file that might not be able to be rendered by the UPnP AV MediaRenderer (e.g. unsupported Media Format Profile for a track URI).

The AVT.CurrentTrackURI virtual instance state variable value is never the URI of the media collection file.

7.4.1.6.11.4

[GUIDELINE] A UPnP AV MediaRenderer that renders a DLNA PlayContainer URI shall use the following mapping for virtual instance state variables.

- The AVT.AVTransportURI value shall be the URI provided in the *CurrentURI* input argument of the AVT:SetAVTransportURI request.
- The AVT.CurrentTrackURI shall be the URI for the track that is currently being rendered. (i.e. the URI for the actual audio-only, audio/video, or image content that is being rendered)

PlayContainerTrackIndex is defined as the index of a rendered content, relative to the sequenced set of content, represented by the DLNA PlayContainer URI.

PlayContainerTotalTracks is defined as the last index of rendered content in the sequenced set of content, represented by the DLNA PlayContainer URI.

PlayContainerTrackIndex and PlayContainerTotalTracks shall be calculated in a manner consistent with these rules.

- Index calculation shall always be done with a preorder traversal of the CDS hierarchy, such that the first CDS object that is played shall have an index of "1". A preorder traversal means that for a given CDS:Browse request (with sort-args applied), left subtrees and leaves (i.e. CDS objects that appear earlier in the CDS:Browse response) are processed before right subtrees and leaves (i.e. CDS objects that appear later in the CDS:Browse response).
- If encountered during traversal, CDS items shall always have a count of "1", regardless of whether they are played or skipped. A skipped CDS item is one where none of the CDS item's <res> elements are rendered by the UPnP AV MediaRenderer. If the CDS item has zero <res> elements, it shall be considered skipped.
- A CDS item with a media collection binary shall always have a count of "1". Furthermore these CDS items shall be skipped. (See 7.4.1.4.28.4 MM DLNA PlayContainer URI for information on the prohibition of playing media collection binaries.)
- If rendered, a CDS item shall automatically render at least one <res> elements associated with the CDS item. If multiple <res> elements are rendered, then they shall be rendered simultaneously. (e.g. A UPnP AV MediaRenderer does not render the same content (in different profiles or transports) multiple times, but a UPnP AV MediaRenderer is permitted to render an image in conjunction with an audio at the same time.)

- If encountered during traversal, a CDS container shall have a count of "0" because <res> elements of a CDS container shall never be played. (See 7.4.1.4.28.4 MM DLNA PlayContainer URI for information on the restriction for playing only CDS items.)

A CDS object shall be counted if and only if there are max-depth-val or fewer CDS containers between the CDS object and the CDS container identified by container-id-val, as defined in 7.4.1.4.28. (i.e. These are the CDS objects that are traversed.)

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [48]	U4ZAU	C

NOTE: In conjunction with other DLNA guidelines that define general rules for ensuring consistency between different AVTransport state variables, this guideline clarifies how those other guidelines apply specifically, in the case of DLNA PlayContainer URI values.

Note that the AVT.NumberOfTracks virtual instance state variable is often updated progressively because it is often difficult to count the number of tracks for all types of media collections, whether they be represented through a DLNA PlayContainer URI or a media collection file.

The AVT.CurrentTrackURI virtual instance state variable value is never the DLNA PlayContainer URI.

7.4.1.6.11.5

[GUIDELINE] A UPnP AV MediaRenderer that renders a media collection file should use the following mapping for virtual instance state variables.

- The AVT.NumberOfTracks value should be the number of content entries in the media collection file (which is often progressively calculated).

[ATTRIBUTES]

S	C	DMR	n/a	n/a	[44] [48]	Q8QS5	

NOTE: Some UPnP AV MediaRenderers might not be able to immediately provide a count of the playback items in a media collection file. The AVTransport specification takes this into account and permits UPnP AV MediaRenderers to update the value in a progressive manner.

This guideline strongly recommends that a UPnP AV MediaRenderer provide the number of playback items in the current media collection file. Sometimes the final value is acquired in an immediate fashion; other times, the value has to be acquired through a counting process. The manner in which the UPnP AV MediaRenderer acquires the total track count is up to the implementer, but the obligation to provide an accurate value exists according to the intentions of the AVTransport specification.

7.4.1.6.11.6

[GUIDELINE] A UPnP AV MediaRenderer that renders a DLNA PlayContainer URI should use the following mapping for virtual instance state variables.

- The AVT.NumberOfTracks value should be the PlayContainerTotalTracks (which is often progressively calculated).

[ATTRIBUTES]

S	C	DMR	n/a	n/a	[44] [48]	SRWQV	

NOTE: Note that the AVT.NumberOfTracks virtual instance state variable is often updated progressively because it is often difficult to count the number of tracks for all types of media collections, whether they be represented through a DLNA PlayContainer URI or a media collection file.

This guideline strongly recommends that a UPnP AV MediaRenderer provide the number of playback items in the current DLNA PlayContainer URI. Sometimes the final value is acquired in an immediate fashion; other times, the value has to be acquired through a counting process. The manner in which the UPnP AV MediaRenderer acquires the total track count is up to the implementer, but the obligation to provide an accurate value exists according to the intentions of the AVTransport specification.

7.4.1.6.11.7

[Guideline] A UPnP AV MediaRenderer that renders a DLNA PlayContainer URI shall use one of the following values for AVT.CurrentTrack virtual instance state variable.

- The value of PlayContainerTrackIndex as defined in 7.4.1.6.11.4.
- A value of “0” to indicate that the PlayContainerTrackIndex is being calculated

[Attributes]

M	C	DMR	n/a	n/a	[44] [48]	Q8ON2	N

NOTE: A UPnP AV MediaRenderer cannot be able to immediately calculate the PlayContainerTrackIndex, as the calculation process could involve multiple interactions with the target UPnP AV MediaServer to traverse the CDS. This guideline states that a UPnP AV MediaRenderer uses a value of “0” for the AVT.CurrentTrack virtual instance state variable to indicate that the correct value of PlayContainerTrackIndex is not yet available.

7.4.1.6.11.8

[Guideline] If no content has been assigned to the UPnP AV MediaRenderer for rendering or the UPnP AV MediaRenderer receives an AVT:SetAVTransportURI action with both the *CurrentURI* and *CurrentURIMetaData* input arguments set to an empty string, then the UPnP AV MediaRenderer shall use the following mapping for virtual instance state variables.

- The AVT.AVTransportURI and AVT.CurrentTrackURI values shall be "".
- The AVT.CurrentTrack and AVT.NumberOfTracks values shall have a value of "0".
- The AVT.TransportState shall have a value of "NO_MEDIA_PRESENT".

[Attributes]

M	C	DMR	n/a	n/a	[44] [48]	VM6GK	

NOTE: This is for the situation after a reset condition for a UPnP AV MediaRenderer or by a UPnP AV MediaRenderer control point through the AVT.SetAVTransportURI action (7.4.1.6.8.2) to clear the rendering state.

7.4.1.6.12 MM GetMediaInfo Behavior

7.4.1.6.12.1

[Guideline] The *CurrentURI* output argument of the AVT:GetMediaInfo action shall return the value of the AVT.AVTransportURI virtual instance state variable.

[Attributes]

M	C	DMR	n/a	n/a	[44]	3YT3F	

NOTE: This requirement repeats what is already stated in the AVTransport specification. Some implementations have accidentally mistaken the *CurrentURI* output argument and the AVT.AVTransportURI values to be always equivalent to the *TrackURI* output argument of AVT:GetPositionInfo and the AVT.CurrentTrackURI virtual instance state variable.

7.4.1.6.12.2

[Guideline] The *NrTracks* output argument of the AVT:GetMediaInfo action shall return the same value as the AVT.NumberOfTracks virtual instance state variable.

[Attributes]

M	R	DMR	n/a	n/a	[44]	OH44W	

NOTE: These guidelines specify the behavior for reporting the value for *NrTracks* (and AVT.NumberOfTracks).

7.4.1.6.12.3

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						GKWSV	

7.4.1.6.12.4

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						YEOC6	

7.4.1.6.12.5

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						4Y37D	

7.4.1.6.12.6

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						4VW4B	

7.4.1.6.12.7

[GUIDELINE] In conjunction with 7.4.1.6.11.3, 7.4.1.6.11.4, 7.4.1.6.11.5, and 7.4.1.6.11.6, a UPnP AV MediaRenderer that progressively updates the value of AVT.NumberOfTracks should provide the total count within 10 seconds of the previous call to AVT:SetAVTransportURI.

[ATTRIBUTES]

S	L	DMR	n/a	n/a	[44]	4Q9J4	

NOTE: DLNA recommends that UPnP AV MediaRenderers provide the count in a timely manner. Control points often display such information to a user and sometimes use the track count information to determine what a user can or cannot do.

This guideline is not required because media collection files can be very large and, in the case of DLNA PlayContainer URIs, CDS hierarchies can be very complex.

7.4.1.6.12.8

[GUIDELINE] The *MediaDuration* output argument of the AVT:GetMediaInfo action shall return the value of the AVT.CurrentMediaDuration virtual instance state variable.

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[44]	44KUE	

7.4.1.6.12.9

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						4KUEX

7.4.1.6.12.10

[GUIDELINE] In conjunction with 7.4.1.6.12.14, a UPnP AV MediaRenderer may progressively update the value of AVT.CurrentMediaDuration, such that the total sum is not provided immediately.

The definition of progressively update is that the UPnP AV MediaRenderer updates the state variable (and appropriately sends GENA events) at a rate that does not exceed one GENA event every 0.2 seconds.

[ATTRIBUTES]

O	R	DMR	n/a	n/a	[44]	Q9J4K

NOTE: Some UPnP AV MediaRenderers might not be able to immediately provide the duration sum for all playback items in a media collection. The AVTransport specification takes this into account and permits UPnP AV MediaRenderers to update the value in a progressive manner.

7.4.1.6.12.11

[GUIDELINE] In conjunction with 7.4.1.6.12.10, a UPnP AV MediaRenderer that progressively updates the value of AVT.CurrentMediaDuration should provide the total sum within 10 seconds of the previous call to AVT:SetAVTransportURI.

[ATTRIBUTES]

S	L	DMR	n/a	n/a	[44]	VW4BW

NOTE: DLNA recommends that UPnP AV MediaRenderers provide the duration sum in a timely manner.

7.4.1.6.12.12

[GUIDELINE] Information returned by AVT:GetMediaInfo should be as accurate as possible. The output arguments that UPnP AV MediaRenderer control points will rely on most will be: *NrTracks*, *CurrentURI*, *CurrentURIMetaData*, and *MediaDuration*.

This also includes the following possibilities and assumptions.

- Some output arguments can be progressively updated as the device acquires more information (e.g., *NrTracks*).
- Some output arguments can have values to indicate that the information cannot be provided (e.g., NOT_IMPLEMENTED value for *MediaDuration* as defined in requirement 7.4.1.6.12.13).
- *CurrentURIMetaData* can have values as defined in guideline requirements 7.4.1.6.14.
- Some output arguments can have placeholder values to indicate information is not yet available. (e.g. DIDL-Lite document with schema-compliant placeholder values)

[ATTRIBUTES]

S	C	DMR	n/a	n/a	[44]	Y37DW

NOTE: This suggestion permits devices to use placeholder values where appropriate-although the guideline strongly encourages information be as accurate as possible.

7.4.1.6.12.13

[GUIDELINE] A UPnP AV MediaRenderer shall do one of the following:

- Never set the value of the AVT.CurrentMediaDuration virtual instance state variable to "NOT_IMPLEMENTED".
- Always set the value of the AVT.CurrentMediaDuration virtual instance state variable to "NOT_IMPLEMENTED".

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [48]	8QS58	

NOTE: UPnP AV MediaRenderers are not allowed to selectively use the string "NOT_IMPLEMENTED" for some contents but not others.

7.4.1.6.12.14

[GUIDELINE] If a UPnP AV MediaRenderer never sets the value of the AVT.CurrentMediaDuration virtual instance state variable to "NOT_IMPLEMENTED", the AVT.CurrentMediaDuration virtual instance state variable should be the total duration for the content specified in AVT.AVTransportURI. When the AVT.AVTransportURI is a media collection or a DLNA PlayContainer URI, the AVT.CurrentMediaDuration virtual instance state variable is the sum of all known playback durations for each item in the media collection or DLNA PlayContainer URI.

[ATTRIBUTES]

S	R	DMR	n/a	n/a	[44]	RWQV7	

NOTE: UPnP AV MediaRenderers often know the playback duration, of items in a media collection or DLNA PlayContainer URI, through summing the durations progressively or by using some metadata within the media collection or through the PlayContainer URI that provides the value in an immediate fashion.

Note that this guideline places no mandatory requirement on the accuracy of the total media duration. If the UPnP AV MediaRenderer is not able to provide a value, then the value of the virtual instance state variable is vendor-dependent.

7.4.1.6.12.15

[GUIDELINE] If a UPnP AV MediaRenderer never sets the value of the AVT.CurrentMediaDuration virtual instance state variable to “NOT_IMPLEMENTED”, but the UPnP AV MediaRenderer cannot compute the duration of the associated content, then the value of state variable shall be set to either “0:00:00” or “00:00:00” to indicate an unknown duration.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44] [48]	VHOQE	N

NOTE: This guideline clarifies that a value of “0:00:00” or “00:00:00” in state variable AVT.CurrentMediaDuration indicates unknown duration for the content associated with AVT.AVTransportURI virtual instance state variable. For example, if the 5th entry in a media collection does not have a duration, then the duration for the entire media collection will be unknown. Another example: If the UPnP AV MediaRenderer is playing an image that is not part of a media collection or a PlayContainer operation, then the value of AVT.CurrentMediaDuration is the same as AVT.CurrentTrackDuration.

7.4.1.6.13 MM GetPositionInfo Behavior

7.4.1.6.13.1

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						EOC6R	

7.4.1.6.13.2

[Guideline] Guideline no longer applies.

[Attributes]

						KWSV3	

7.4.1.6.13.3

[Guideline] The AVT:GetPositionInfo shall return values in the following manner.

- The *Track* output argument shall be equal to AVT.CurrentTrack value
- The *TrackURI* output argument shall be equal to AVT.CurrentTrackURI value

Note that there can be a time period where the URI and track values are temporarily non-synchronized for transitioning purposes.

[Attributes]

M	R	DMR	n/a	n/a	[44]	A68VL	

NOTE: When the UPnP AV MediaRenderer is in the TRANSITIONING state, the URI values and track index values can be non-synchronized. When the UPnP AV MediaRenderer enters a non-transitional state (e.g. PLAYING, STOPPED, etc.), the URI and track values are expected to be accurate as described in the guideline.

7.4.1.6.13.4

[Guideline] Guideline no longer applies.

[Attributes]

						H44WM	

7.4.1.6.13.5

[Guideline] Guideline no longer applies.

[Attributes]

						YT3FQ	

7.4.1.6.13.6

[Guideline] Guideline no longer applies.

[Attributes]

						2HVW3	

7.4.1.6.13.7

[Guideline] Information returned by AVT:GetPositionInfo should be as accurate as possible. The output arguments that control points will rely on most will be: *RelTime*, *TrackDuration*, *Track*, and *TrackURI*.

[Attributes]

S	C	DMR	n/a	n/a	[44]	74MVR	

NOTE: This suggestion permits devices to use placeholder values where appropriate-although the guideline strongly encourages information be as accurate as possible.

7.4.1.6.13.8

[GUIDELINE] The *TrackDuration* output argument of the AVT:GetPositionInfo action shall return the value of the AVT.CurrentTrackDuration virtual instance state variable.

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[44] [48]	XTD3B	

7.4.1.6.13.9

[GUIDELINE] A UPnP AV MediaRenderer shall do one of the following:

- Never set the value of the AVT.CurrentTrackDuration virtual instance state variable to "NOT_IMPLEMENTED".
- Always set the value of the AVT. CurrentTrackDuration virtual instance state variable to "NOT_IMPLEMENTED".

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [48]	59RRB	

NOTE: UPnP AV MediaRenderers are not allowed to selectively use the string "NOT_IMPLEMENTED" for some tracks but not others.

7.4.1.6.13.10

[GUIDELINE] If a UPnP AV MediaRenderer never sets the value of the AVT.CurrentTrackDuration virtual instance state variable to "NOT_IMPLEMENTED", then the AVT.CurrentTrackDuration virtual instance state variable should be the duration for the track specified in AVT.CurrentTrackURI.

[ATTRIBUTES]

S	R	DMR	n/a	n/a	[44]	8EII9	

NOTE: Provides baseline expectations for reporting the playback duration of the track that is currently being rendered. This guideline does not impose any mandatory accuracy requirements because methodologies for determining the playback duration varies between implementations and often has a dependency on the media formats. If the UPnP AV MediaRenderer is not able to provide a value, then the value of the virtual instance state variable is vendor-dependent.

7.4.1.6.13.11

[GUIDELINE] If a UPnP AV MediaRenderer never sets the value of the AVT.CurrentTrackDuration virtual instance state variable to "NOT_IMPLEMENTED", but the UPnP AV MediaRenderer cannot compute the duration of the track (for example, if the track is an image), then the value of this state variable shall be set to either "0:00:00" or "00:00:00" to indicate an unknown duration.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44] [48]	Y6LY5	N

NOTE: This guideline clarifies that a value of "0:00:00" or "00:00:00" in state variable AVT.CurrentTrackDuration indicates unknown track duration for the current media resource.

7.4.1.6.13.12

[GUIDELINE] The RelTime output argument of the AVT:GetPositionInfo action shall return the value of the AVT.RelativeTimePosition virtual instance state variable.

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[44] [48]	P5VR2	

7.4.1.6.13.13

[GUIDELINE] A UPnP AV MediaRenderer shall do one of the following:

- Never set the value of the AVT.RelativeTimePosition virtual instance state variable to "NOT_IMPLEMENTED".
- Always set the value of the AVT.RelativeTimePosition virtual instance state variable to "NOT_IMPLEMENTED".

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [48]	TD3B8	

NOTE: UPnP AV MediaRenderers are not allowed to selectively use the string "NOT_IMPLEMENTED" for some tracks but not others.

7.4.1.6.13.14

[GUIDELINE] If a UPnP AV MediaRenderer never sets the value of the AVT.RelativeTimePosition virtual instance state variable to "NOT_IMPLEMENTED", the AVT.RelativeTimePosition virtual instance state variable shall indicate the current playback position in terms of time for the content indicated by AVT.CurrentTrackURI. The value of the AVT.RelativeTimePosition virtual instance state variable shall not be "END_OF_MEDIA".

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[44]	9RRBW	

NOTE: This guideline specifies the correct behavior for reporting the *RelTime* output arguments. Reporting an empty string is not acceptable.

Notice also that reporting "END_OF_MEDIA" is not acceptable in the DLNA context.

DLNA only defines the usage of AVT.RelativeTimePosition. The use of AVT.AbsoluteTimePosition and the *AbsTime* output argument of the AVT:GetPositionInfo action is vendor dependent.

7.4.1.6.13.15

[GUIDELINE] The *TrackMetaData* output argument of the AVT:GetPositionInfo action shall return the value of the AVT.CurrentTrackMetaData virtual instance state variable.

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[44] [48]	EII9O	

7.4.1.6.13.16

[GUIDELINE] A UPnP AV MediaRenderer shall do one of the following:

- Never set the value of the AVT.CurrentTrackMetaData virtual instance state variable to "NOT_IMPLEMENTED".
- Always set the value of the AVT.CurrentTrackMetaData virtual instance state variable to "NOT_IMPLEMENTED".

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [48]	5VR2J	

NOTE: UPnP AV MediaRenderers are not allowed to selectively use the string "NOT_IMPLEMENTED" for some tracks but not others.

7.4.1.6.13.17

[GUIDELINE] If a UPnP AV MediaRenderer never sets the value of the AVT.CurrentTrackMetaData virtual instance state variable to "NOT_IMPLEMENTED", the AVT.CurrentTrackMetaData virtual instance state variable shall be formatted in one of the following ways:

- The value shall specify a valid DIDL-Lite XML fragment as defined in 7.4.1.6.14.9, with a single <item> element that describes the track indicated by the AVT.CurrentTrackURI instance state variable.
- The value shall be an empty string when no metadata is available for the current track indicated by the AVT.CurrentTrackURI virtual instance state variable, or when the AVT.CurrentTrackURI virtual instance state variable is also an empty string (e.g. no content is set up for rendering).

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[44]	D3B8T

7.4.1.6.14 MM Metadata Reporting

7.4.1.6.14.1

[GUIDELINE] If a UPnP AV MediaRenderer always sets the value of the AVT.AVTransportURIMetaData virtual instance state variable to "NOT_IMPLEMENTED", then the UPnP AV MediaRenderer shall accept an AVT:SetAVTransportURI request as though an empty value was sent for the *CurrentURIMetaData*.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44]	4MVRM

NOTE: This guideline essentially requires UPnP AV MediaRenderers to discard the *CurrentURIMetaData* value if the input argument is not supported. This guideline allows control points to use the *CurrentURIMetaData* input argument when invoking AVT:SetAVTransportURI, without having to implement logic for retrying the request without metadata.

Note that requirement 7.3.2.15.1 only requires UPnP devices to accept SOAP requests up to 20,480 bytes (20 KB) in size.

7.4.1.6.14.2

[GUIDELINE] If a UPnP AV MediaRenderer never sets the value of the AVT.AVTransportURIMetaData virtual instance state variable to "NOT_IMPLEMENTED", then the UPnP AV MediaRenderer may choose not to validate the *CurrentURIMetaData* input argument specified by the UPnP AV MediaRenderer control point in the AVT:SetAVTransportURI request.

[ATTRIBUTES]

O	C	DMR	n/a	n/a	[44]	HVW39

NOTE: This guideline absolves UPnP AV MediaRenderers from having to parse or validate the DIDL-Lite metadata sent by a control point.

7.4.1.6.14.3

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						ZAU8B

7.4.1.6.14.4

[GUIDELINE] The value of the *CurrentURIMetaData* output argument for the AVT:GetMediaInfo action shall be the same as the value of the AVT.AVTransportURIMetaData virtual instance state variable.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44]	T3FQL	

7.4.1.6.14.5

[GUIDELINE] A UPnP AV MediaRenderer shall update the AVT.AVTransportURI virtual instance state variable to indicate the current URI prior to changing the AVT.TransportState virtual instance state variable to "PLAYING".

[ATTRIBUTES]

M	L	DMR	n/a	n/a	[44]	44WMJ	

NOTE: This allows control points to know when the UPnP AV MediaRenderer has selected a different <res> element.

Control points can rely on the GENA event to indicate when the UPnP AV Media Renderer has chosen an alternate URI.

This guideline applies generally, including these scenarios:

- The UPnP AV MediaRenderer uses the URI specified in the *CurrentURI* argument of the AVT:SetAVTransportURI request.
- The DMR overrides the specified URI (in the *CurrentURI* argument) with another one (from the *CurrentURIMetaData* argument) in the AVT:SetAVTransportURI request (as described in 7.4.1.6.9.1).

7.4.1.6.14.6

[GUIDELINE] UPnP AV MediaRenderers may specify a value for the AVT.AVTransportURIMetaData virtual instance state variable (and hence the *CurrentURIMetaData* output argument of AVT:GetMediaInfo) that is different from the value of the *CurrentURIMetaData* argument of the last AVT:SetAVTransportURI request.

[ATTRIBUTES]

O	C	DMR	n/a	n/a	[44]	68VLX	

NOTE: UPnP AV MediaRenderers can have different values because of a variety of reasons.

- The device can parse different metadata from within the content file.
- The device can remove elements and attributes from the DIDL-Lite XML fragment (while remaining schema-compliant) to reduce memory use.
- The device can truncate values of elements and attributes to reduce memory use.

7.4.1.6.14.7

[GUIDELINE] If a UPnP AV MediaRenderer never sets the value of the AVT.AVTransportURIMetaData virtual instance state variable to "NOT_IMPLEMENTED" and it specifies a value for the AVT.AVTransportURIMetaData virtual instance state variable (and hence the *CurrentURIMetaData* output argument of AVT:GetMediaInfo) that is different from the value of the *CurrentURIMetaData* argument of the last AVT:SetAVTransportURI request, then the UPnP AV MediaRenderer shall impose the following restrictions on the metadata.

- The provided metadata shall represent the metadata of the content indicated by the AVT.AVTransportURI virtual instance state variable.

- If the AVT.AVTransportURI virtual instance state variable points to a media collection, then the provided metadata shall be limited to the media collection. The provided metadata shall not include metadata for items within the media collection.
- The provided metadata shall specify a valid DIDL-Lite XML fragment as defined in 7.4.1.6.14.9.

[ATTRIBUTES]

M	L	DMR	n/a	n/a	[44]	WSV39

NOTE: Many control points have problems accepting large metadata values. This guideline limits the metadata for AVT.AVTransportURIMetaData and *CurrentURIMetaData* to refer specifically to the content indicated by the AVT.AVTransportURI virtual instance state variable. This guideline prohibits the general practice of collectively using the metadata of each item in a media collection to represent the metadata of a media collection.

7.4.1.6.14.8

[GUIDELINE] If a UPnP AV MediaRenderer is capable of playing a media collection and if the UPnP AV MediaRenderer has access to metadata of individual items in the collection, then the UPnP AV MediaRenderer should report that metadata through the AVT.CurrentTrackMetaData virtual instance state variable.

[ATTRIBUTES]

S	C	DMR	n/a	n/a	[44]	OC6RF

NOTE: Control points often use metadata provided by the UPnP AV MediaRenderer in user interfaces. UPnP AV MediaRenderers are strongly encouraged to provide such metadata whenever possible.

7.4.1.6.14.9

[GUIDELINE] If a UPnP AV MediaRenderer control point specifies a value for the *CurrentURIMetaData* argument of an AVT:SetAVTransportURI request, then the control point shall follow these restrictions for the value of the *CurrentURIMetaData* argument.

- compliant with the DIDL-Lite schema
- exactly one <DIDL-Lite> element
- exactly one <item> or <container> element
- exactly one <dc:title> element and value
- a minimum of zero and a maximum of one <dc:creator> element and value
- exactly one <upnp:class> element and value
- a minimum of one <res> element

All other XML elements are permitted as long as they are properly declared with their namespaces.

The provided metadata shall represent the metadata of the content indicated by the *CurrentURI* input argument.

One of the <res> elements shall be the <res> element that contains the URI specified in the *CurrentURI* input argument.

If the *CurrentURI* input argument points to a media collection, then the provided metadata shall be limited to the media collection. The provided metadata shall not include metadata for items within the media collection.

[ATTRIBUTES]

M	L	DMC +PU+	M-DMC	MIU	[44]	37DW6

NOTE: This guideline limits the scope of the *CurrentURIMetaData* to the metadata directly associated with the *CurrentURI* input argument. Metadata for a media collection is permitted, so long as value does not provide metadata for each of the individual items in the collection.

Many UPnP AV MediaRenderers have problems storing a large amount of metadata provided in the *CurrentURIMetaData*. Equally problematic is the fact that many control points cannot support a scenario where a UPnP device transmits a UPnP event or an AVT:GetMediaInfo response with a large amount of metadata.

The expected metadata to be sent in the *CurrentURIMetaData* argument is best described by the CDS:Browse response for the following request.

- *ObjectID*: The CDS object ID of that provided the URI specified in the *CurrentURI* input argument of AVT:SetAVTransportURI.
- BrowseFlag: BrowseMetadata
- Filter: One or more of the following: ALLIP, res (and or any res attribute), dc:creator, and any other metadata that the control point wants to provide

Whenever possible, control points are encouraged to provide all of the available <res> attributes that are normative for DIDL-Lite. Likewise, UPnP AV MediaRenderers are encouraged to accept and preserve these attributes.

Lastly the guideline permits control points to specify a single <res> element in the metadata, on behalf of a user request. In such cases, the *CurrentURIMetaData* argument only includes the <res> element that corresponds to the URI specified in the *CurrentURI* argument.

However, control points are encouraged to provide all available <res> elements. This allows UPnP AV MediaRenderers the opportunity to choose a <res> element that might provide a better rendering experience. See 7.4.1.6.9 for more information about UPnP AV MediaRenderers that select alternate URIs from the *CurrentURIMetaData* argument.

7.4.1.6.14.10

[GUIDELINE] UPnP AV MediaRenderer control points that receive metadata from a UPnP AV MediaRenderer shall be tolerant of DIDL-Lite metadata that is valid and conformant to DLNA restrictions.

Tolerant behavior is defined as being able to parse-and-accept or parse-and-ignore the metadata. Failing to parse a DLNA-compliant UPnP action response or event because of metadata is unacceptable behavior.

[ATTRIBUTES]

M	C	DMC +PU+	M-DMC	MIU	[44]	W4BWN

NOTE: Control points that invoke UPnP actions or subscribe to UPnP events that involve metadata needs to be prepared for the presence of metadata, even if metadata is not always provided.

7.4.1.6.14.11

[GUIDELINE] UPnP AV MediaRenderer control points that receive metadata from a UPnP AV MediaRenderer should be tolerant of DIDL-Lite metadata that is invalid or not conformant to DLNA restrictions.

Tolerant behavior is defined as being able to parse-and-ignore the metadata.

[ATTRIBUTES]

S	C	DMC +PU+	M-DMC	MIU	[44]	9J4K4

NOTE: UPnP AV MediaRenderer control points that can handle DIDL-Lite metadata that is not schema compliant exhibit a good level of robustness in an environment where DLNA UPnP AV MediaRenderer control points can be interacting with non-DLNA UPnP AV MediaRenderer devices.

7.4.1.6.14.12

[GUIDELINE] The UPnP AV MediaRenderer shall do one of the following:

- Never set the value of the AVT.AVTransportURIMetaData virtual instance state variable to "NOT_IMPLEMENTED".
- Always set the value of the AVT.AVTransportURIMetaData virtual instance state variable to "NOT_IMPLEMENTED".

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44]	RRBWY

NOTE: UPnP AV MediaRenderers are not allowed to selectively use the string "NOT_IMPLEMENTED" for some contents but not others.

7.4.1.6.14.13

[GUIDELINE] If a UPnP AV MediaRenderer never sets the value of the AVT.AVTransportURIMetaData virtual instance state variable to "NOT_IMPLEMENTED", then the AVT.AVTransportURIMetaData virtual instance state variable shall be formatted in one of the following ways:

- The value shall specify a valid DIDL-Lite XML fragment as defined in 7.4.1.6.14.9, with a single <item> element that describes the content indicated by the AVT.AVTransportURI virtual instance state variable.
- The value shall be an empty string when no metadata is available for the content indicated by the AVT.AVTransportURI virtual instance state variable, or when the AVT.AVTransportURI virtual instance state variable is also an empty string (e.g. no content is set up for rendering) or a DLNA PlayContainer URI.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44]	VR2JS

NOTE: The UPnP AV MediaRenderer can obtain the metadata from the *CurrentURIMetaData* input argument in the AVT:SetAVTransportURI action, or by using other means (see also 7.4.1.6.14.6).

7.4.1.6.15 MM Reporting Transport Information

7.4.1.6.15.1

[GUIDELINE] UPnP MediaRenderers that respond to an AVT:GetTransportInfo request shall reflect the play/transport state in the following manner.

- The *CurrentTransportState* output argument shall match the AVT.TransportState instance state variable.
- The *CurrentTransportStatus* output argument shall match the AVT.TransportStatus instance state variable.
- The *CurrentSpeed* output argument shall match the AVT.TransportPlaySpeed instance state variable.

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[44]	KUEXL

NOTE: This guideline makes it so that control points can safely assume that UPnP actions and instance state variables report the same transport state information.

7.4.1.6.15.2

[GUIDELINE] UPnP MediaRenderers that respond to an AVT:GetTransportSettings request shall accurately reflect the *PlayMode* output argument to match the AVT.CurrentPlayMode instance state variable.

[Attributes]

M	R	DMR	n/a	n/a	[44]	UEXLS	

7.4.1.6.16 MM Normative MediaRenderer State Transitions

7.4.1.6.16.1

[Guideline] If a UPnP MediaRenderer enters the TRANSITIONING state, it shall change to the state desired by the control point within 30 seconds.

The longest period of time that a MediaRenderer device is permitted to remain in the TRANSITIONING state is 30 seconds.

[Attributes]

M	L	DMR	n/a	n/a	[44]	J4K4Z	

NOTE: The TRANSITIONING state is a way for a device to indicate that it is attempting to change into a different state, such as PLAYING or STOPPED.

7.4.1.6.16.2

[Guideline] UPnP MediaRenderers may enter the TRANSITIONING state at any time.

[Attributes]

O	C	DMR	n/a	n/a	[44]	4BWN9	

NOTE: The AVTransport specification has an informative diagram that describes TRANSITIONING to be used only between STOPPED and PLAYING states. This diagram is not restrictive, and it allows new transitions.

The TRANSITIONING provides a useful cue to the user that the device is trying to do something. For example, entering the TRANSITIONING state after a call to AVT:SetAVTransportURI acknowledges the user's request to change content even if playback has not yet begun. Likewise, if a device is in the PLAYING state and network problems interrupt playback, the device can go into the TRANSITIONING state during the interruption.

7.4.1.6.16.3

[Guideline] UPnP MediaRenderers shall not define a new intermediate state.

An intermediate state is a state that the device enters temporarily before entering the state requested by the control point.

[Attributes]

M	L	DMR	n/a	n/a	[44]	7DW6X	

NOTE: Defining a new intermediate state can confuse control points into believing other control points are trying to use the device. A control point that encounters a vendor-defined state of BUFFERING has no idea that this state is an intermediate state. Likewise, a control point that encounters a vendor-defined state of LOCKED has no idea that the device might remain in this state indefinitely. This guideline allows control points to always assume that vendor-defined states can last indefinitely and that TRANSITIONING is the only intermediate state.

7.4.1.6.16.4

[Guideline] UPnP MediaRenderers may define new non-intermediate states.

[Attributes]

O	L	DMR	n/a	n/a	[44]	C6RFR	

NOTE: For example, defining a LOCKED state to indicate that the DMR cannot accept AVTransport requests is acceptable due to its current state (e.g. a local user is using the device directly).

However, creating a new state called BUFFERING to represent that the device is busy buffering data in preparation for rendering is not permitted.

Note that the LOCKED and BUFFERING states are only examples. The key distinction between the examples is the former involves an out-of-scope scenario and the latter involves an in-scope scenario. In the case of LOCKED, an external stimulus (i.e. out-of-scope scenario) caused the device to enter the LOCKED state. In the case of BUFFERING, the DMR entered the BUFFERING state after a control point requested a normative state change request (e.g. play state change, track change, URI change, etc.).

7.4.1.6.16.5

[GUIDELINE] UPnP MediaRenderers may define new allowed values for the AVT.TransportStatus instance state variable.

[ATTRIBUTES]

O	C	DMR	n/a	n/a	[44]	SV393	

NOTE: In lieu of defining new intermediate states, vendors are permitted to use the TransportStatus instance state variable to convey additional information about the transport layer.

7.4.1.6.16.6

[GUIDELINE] UPnP MediaRenderers that define new allowed values for the AVT.TransportStatus instance state variable should begin the allowed value with "ERROR_" to indicate the value represents an error condition.

[ATTRIBUTES]

S	L	DMR	n/a	n/a	[44]	8VLXY	

NOTE: This guideline allows control points to mark status values as being informative or error-related. The normative error value for AVT.TransportStatus is ERROR_OCCURRED.

7.4.1.6.16.7

[GUIDELINE] If a UPnP AV MediaRenderer responds to an AVT:Seek action with the 200 (OK) response code and the value of the AVT.TransportState virtual instance state variable is PAUSED_PLAYBACK or STOPPED, then it shall set the AVT.TransportState virtual instance state variable to a value of TRANSITIONING. After the seek operation is completed (i.e. the desired playback position is reached), the UPnP AV MediaRenderer shall set the AVT.TransportState virtual instance state variable to the value before the transition to TRANSITIONING.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44]	76CN7	N

NOTE: This guideline allows UPnP AV MediaRenderer control points to detect a change to the transport state (AVT.TransportState) on the UPnP AV MediaRenderer through the eventing of the AVT.LastChange virtual state variable and could be used to trigger an invocation of the AVT.GetPositionInfo action. This guideline applies even when performing an instantaneous seek on cached content and remaining in the same transport state.

7.4.1.6.16.8

[GUIDELINE] A UPnP AV MediaRenderer control point should always invoke the AVT:GetPositionInfo action when the AVT.LastChange evented state variable contains an update to the AVT.TransportState virtual instance state variable with a value of PAUSED_PLAYBACK or STOPPED even when the AVT.TransportState virtual instance state variable value is the same as currently cached in the UPnP AV MediaRenderer control point.

[ATTRIBUTES]

S	A	DMC +PU+	M-DMC	n/a	[44]	VQ7SL	N

NOTE: This guideline gives guidance to UPnP AV MediaRenderer control points to refresh their current position information when a UPnP AV MediaRenderer reports a transport state change to PAUSED_PLAYBACK or STOPPED. For example, when a UPnP AV MediaRenderer control point, not actively controlling a UPnP AV MediaRenderer, observes over the network that a UPnP AV MediaRenderer is currently in the PAUSED_PLAYBACK or STOPPED state, then seeks to a new position where it enters the TRANSITIONING state, and then transitions back to the PAUSED_PLAYBACK or STOPPED state when the seek completes, the UPnP AV MediaRenderer control point needs to invoke an AVT:GetPositionInfo action to the UPnP AV MediaRenderer after each transport state change including when the seek completes (i.e. returns to the PAUSED_PLAYBACK or STOPPED state) to refresh the current position.

7.4.1.6.16.9

[GUIDELINE] If a UPnP AV MediaRenderer receives an AVT.Play request to play an image, then it shall transition into a "PLAYING" state as soon as it successfully decodes and starts rendering the image.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [48]	6GN3G	N

NOTE: This guideline together with 7.4.1.6.16.10 and 7.4.1.6.16.11 define the use of the "PLAYING" state for UPnP AV MediaRenderers that play image content.

7.4.1.6.16.10

[GUIDELINE] If a UPnP AV MediaRenderer is currently playing an image, it shall remain in the "PLAYING" state until it receives a new action that changes its state, or until some third-party application forces the UPnP AV MediaRenderer to adopt a different state using means outside the scope of DLNA.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [48]	CFWQ5	N

NOTE: This guideline together with 7.4.1.6.16.7 and 7.4.1.6.16.11 define the use of the "PLAYING" state for UPnP AV MediaRenderers that play image content.

For example, if a TV that is operating as a UPnP AV MediaRenderer receives a request to display a picture, the TV will change its state into "PLAYING" as soon as it starts displaying the picture on the screen. The TV will remain in this state, and will remain displaying the picture, until one of two things happen: (1) The TV receives a request to change its state from a UPnP AV MediaRenderer Control Point in the network. The request source could be the original UPnP AV MediaRenderer Control Point or a different one. (2) A different application forces the TV to change its state. Examples of the latter are: a user tunes to some channel for watching TV, or the TV starts a screen saver application after some time of inactivity.

7.4.1.6.16.11

[GUIDELINE] If a UPnP AV MediaRenderer that has been playing an image stops displaying the image due to the intervention of a third-party application, it shall enter into the 'STOPPED" or "NO_MEDIA_PRESENT" states.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [48]	93BWN	N

NOTE: This guideline together with 7.4.1.6.16.7 and 7.4.1.6.16.10 define the use of the "PLAYING" state for UPnP AV MediaRenderers that play image content.

For example, if a TV operating as a UPnP AV MediaRenderer is displaying a picture and changes the picture due to some screen saver application, the TV will change its state into "STOPPED" or "NO_MEDIA_PRESENT" as long as the TV can still respond successfully to requests to play media from the network. Notice that some third-party applications do not allow the TV to receive control actions as long as the third-party application is controlling the device. For example, a TV that is currently used for watching broadcast channels could block action requests from the network to play media.

7.4.1.6.16.12

[GUIDELINE] If a UPnP AV MediaRenderer is currently playing an image, and it receives an AVT:Stop action, then it shall transition into the “STOPPED” state.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [48]	GWEVN	N

7.4.1.6.16.13

[GUIDELINE] If in a UPnP AV MediaRenderer the value of the AVT.CurrentTrackURI virtual instance state variable is an image URI, and if the UPnP AV MediaRenderer is in the “STOPPED” state, then the UPnP AV MediaRenderer shall not display the image.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [48]	FAINZ	N

NOTE: This guideline clarifies that when a UPnP AV MediaRenderer is in the “STOPPED” state, and the currently available content is an image, the UPnP AV MediaRenderer needs to clear the screen and not show the image.

7.4.1.6.17 MM Transport Actions

7.4.1.6.17.1

[GUIDELINE] The comma-separated list of values listed in the AVT.CurrentTransportActions virtual instance state variable may change depending on what the device is doing and what content the device is accessing.

[ATTRIBUTES]

O	R	DMR	n/a	n/a	[44]	VLXYX	

NOTE: This guideline requirement references the permitted and expected behaviors of a device that dynamically enables/disables its AVTransport actions.

7.4.1.6.17.2

[GUIDELINE] The value returned in the *Actions* output argument of an AVT:GetCurrentTransportActions request shall match the value of the AVT.CurrentTransportActions virtual instance state variable.

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[44]	V393O	

7.4.1.6.17.3

[GUIDELINE] A UPnP AV MediaRenderer control point should provide a UI indicator to inform a user of disabled transport actions.

[ATTRIBUTES]

S	C	DMC +PU+	M-DMC	n/a	n/a	6RFRG	

NOTE: This guideline requires control points to provide UI indications to inform users that a playback feature is disabled. Without this information, a user can be misled into believing that the device is not working properly. Examples of user indicators include the following:

- gray-out the button for the disabled operation.
- an icon that flashes on the screen to indicated the disabled state when the user pushes the button.

Depending on UI form factor this guideline can be difficult to implement, such as a handheld remote with physical buttons.

7.4.1.6.17.4

[Guideline] A UPnP AV MediaRenderer shall report the disabling of the Pause operation by excluding the value “Pause” from the *Actions* output argument of an AVT:GetCurrentTransportActions and the AVT.CurrentTransportActions virtual instance state variable.

[Attributes]

M	C	DMR	n/a	n/a	n/a	DW6X6	

7.4.1.6.17.5

[Guideline] If a UPnP AV MediaRenderer's AVT.NumberOfTracks value is greater than 1, then the AVT:Next/AVT:Previous actions shall have the behavior of incrementing/decrementing the AVT:CurrentTrack virtual instance state variable (and likewise properly reflecting other state change information required by the AVTransport specification).

[Attributes]

M	R	DMR	n/a	n/a	[44]	BWN9C	

NOTE: This guideline defines the behavior of AVT:Next and AVT:Previous in the context of playing a media collection. Behavior for AVT:Next and AVT:Previous is not defined for other cases.

7.4.1.6.17.6

[Guideline] If a UPnP AV MediaRenderer's AVT.NumberOfTracks value is equal to 1, then a UPnP AV MediaRenderer control point should not issue the AVT:Next and AVT:Previous actions, as the behavior performed by a UPnP AV MediaRenderer is implementation dependent.

[Attributes]

S	C	DMC +PU+	M-DMC	n/a	[44]	3B8T5	

NOTE: The behavior is not defined in [44] and preferably will not be used in this particular situation.

7.4.1.6.17.7

[Guideline] A UPnP AV MediaRenderer shall implement minimally the values "None" and "Network" for the AVT.PossiblePlaybackStorageMedia virtual instance state variable and the PlayMedia output parameter for the AVT:GetDeviceCapabilities action.

[Attributes]

						II9O4	D

NOTE: This is the minimal requirement to achieve the DLNA System Usages.

7.4.1.6.17.8

[Guideline] If in a UPnP AV MediaRenderer, the value of the AVT.TransportURI virtual instance state variable is an image URI, then the value of the AVT.CurrentTransportActions virtual instance state variable should not include the value “Pause”.

[Attributes]

S	C	DMR	n/a	n/a	[44] [48]	B2JKO	N

NOTE: If a UPnP AV MediaRenderer plays an image, the relevant transport actions are “Play” and “Stop. The UPnP AV MediaRenderer could also accept “Pause”, but the behavior is vendor-dependent.

7.4.1.6.17.9

[GUIDELINE] If in a UPnP AV MediaRenderer, the value of the AVT.TransportURI virtual instance state variable is a DLNA PlayContainer URI, or a URI of a media collection file, and the AVT.CurrentTrackURI virtual instance state variable is an image URI, then the value of AVT.CurrentTransportActions virtual instance state variable may include the value "Pause".

[ATTRIBUTES]

O	C	DMR	n/a	n/a	[44] [48]	B5DWR	N

NOTE: If a UPnP AV MediaRenderer plays an image within a playlist, it could accept "Pause". The behavior is described in 7.4.1.6.17.10.

7.4.1.6.17.10

[GUIDELINE] A If in a UPnP AV MediaRenderer the value of the AVT.CurrentTrackURI virtual instance state variable is an image URI, and if the UPnP AV MediaRenderer lists "Pause" in the AVT.CurrentTransportActions virtual instance state variable, then upon receiving an AVT:Pause action, the UPnP AV MediaRenderer shall transition into the "PAUSED_PLAYBACK" state.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44] [48]	MFW38	N

NOTE: Some UPnP AV MediaRenderers will support Pause as a transport operation available when the current track is an image. This transport action is useful only when the AVT.AVTransportURI virtual instance state variable contains the URI of a media collection file or is a DLNA PlayContainer URI.

7.4.1.6.17.11

[GUIDELINE] If in a UPnP AV MediaRenderer the value of the AVT.CurrentTrackURI virtual instance state variable is an image URI, and if the UPnP AV MediaRenderer is in the "PAUSED_PLAYBACK" state, then the UPnP AV MediaRenderer may display the image.

[ATTRIBUTES]

O	C	DMR	n/a	n/a	[44] [48]	O9GZ4	N

NOTE: This guideline clarifies that when a UPnP AV MediaRenderer is in the "PAUSED_PLAYBACK" state, and the current track is an image, the UPnP AV MediaRenderer may continue displaying the image.

7.4.1.6.18 MM Play Mode Behavior

7.4.1.6.18.1

[GUIDELINE] UPnP MediaRenderers that implement AVT:SetPlayMode shall implement the method such that changes to the current play mode are applied immediately.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44]	4K4ZY	

NOTE: An example of bad behavior is a MediaRenderer that requires a control point to invoke AVT:Play after a call to AVT:SetPlayMode in order for the requested play mode to be applied.

7.4.1.6.18.2

[GUIDELINE] UPnP MediaRenders that change the play mode may change the transport state so long as the new state is not STOPPED.

[ATTRIBUTES]

O	C	DMR	n/a	n/a	[44]	EXLSF	

7.4.1.6.18.3

[GUIDELINE] UPnP MediaRenderers should keep the same play mode after a call to AVT:SetAVTransportURI.

[ATTRIBUTES]

S	C	DMR	n/a	n/a	[44]	XLSFB	

NOTE: Automatically changing unrelated portion of the device's state is not a good practice.

7.4.1.6.19 MM Play Modes

7.4.1.6.19.1

[GUIDELINE] UPnP MediaRenderers that implement the "NORMAL" play mode shall implement it the following manner.

- An AVT:Next request results in AVT.CurrentTrack being incremented by one.
- AVT:Next requests that attempt to change the track number beyond the last track shall result with no state change (i.e. request accepted and ignored) or a response with a UPnP AV error code 711 (Illegal seek target).
- An AVT:Previous request results in AVT.CurrentTrack being decremented by one
- AVT:Previous requests that attempt to change the track number before the first track shall result with no state change (i.e. request accepted and ignored) or a UPnP AV error code 711 (Illegal seek target).
- If a new value is applied to AVT.CurrentTrack, then AVT.CurrentTrackURI is updated appropriately.
- If the play state before an AVT:Next or AVT:Previous request is PLAYING and a new track is applied, then the device continues playback with the new track.
- If the device is in the PLAYING state, with AVT.CurrentTrack = AVT.NumberOfTracks, and playback finishes for the content indicated by AVT.CurrentTrackURI, then the MediaRenderer enters the STOPPED state and AVT.CurrentTrack is reset to 1. AVT.CurrentTrackURI is appropriately updated, but AVT.AVTransportURI remains the same. This bulleted item does not apply wihen AVT.NextAVTransportURI is set as a result of AVT:SetNextAVTransprotURI.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44]	K4ZYS	

NOTE: The AVTransport specification does not specify the behavior of different play modes. These guidelines specify the basic expectations, inspired largely from traditional consumer electronics devices that play optical media content. The AVTransport specification can require additional behaviors of the MediaRenderer.

7.4.1.6.19.2

[GUIDELINE] UPnP MediaRenderers that implement the "REPEAT_ONE" play mode shall implement it in the same manner as 7.4.1.6.19.1, except in the following manners.

- If the device is in the PLAYING state and playback reaches the end for the content indicated by AVT.CurrentTrackURI, then the MediaRenderer changes the playback position to "00:00:00" and continues playing the same content from the beginning.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44]	WN9C8	

7.4.1.6.19.3

[GUIDELINE] UPnP MediaRenderers that implement the "REPEAT_ALL" play mode shall implement it in the same manner as 7.4.1.6.19.1, with the following exceptions.

- AVT:Next requests that attempt to change the track number beyond the last track shall result with AVT.CurrentTrack set to "1".
- AVT:Previous requests that attempt to change the track number before the first track shall result with AVT.CurrentTrack set to AVT.NumberOfTracks.
- If the device is in the PLAYING state, with AVT.CurrentTrack = AVT.NumberOfTracks, and playback finishes for the content indicated by AVT.CurrentTrackURI, then the MediaRenderer resets AVT.CurrentTrack to 1 and AVT.CurrentTrackURI is appropriately updated. AVT.AVTransportURI remains unchanged and content playback continues with the new track.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44]	W6X6H	

7.4.1.6.19.4

[GUIDELINE] UPnP MediaRenderers that implement the "RANDOM" play mode shall implement it in the following manner.

- AVT:Next and AVT:Previous requests shall result with AVT.CurrentTrack set to a random value from "1" to AVT.NumberOfTracks, with the AVT.CurrentTrackURI getting updated appropriately. If the play state is PLAYING, then content playback continues with the new track.
- If a new value is applied to AVT.CurrentTrack, then AVT.CurrentTrackURI getting updated appropriately.
- If the play state before an AVT:Next or AVT:Previous request is PLAYING and a new track is applied, then the device continues playback with the new track.
- If the device is in the PLAYING state and playback finishes for the content indicated by AVT.CurrentTrackURI, then the MediaRenderer sets a new random value for AVT.CurrentTrack and AVT.CurrentTrackURI is appropriately updated. AVT.AVTransportURI remains unchanged and content playback continues with the new track.

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44]	RFRGK	

7.4.1.6.19.5

[GUIDELINE] UPnP MediaRenderers that implement the "SHUFFLE" play mode shall implement it in the same manner as "RANDOM" (see 7.4.1.6.19.4) except for the following manners.

- The device shall track the value history of AVT.CurrentTrack so that the new track value is not a repeat of a previously played track.
- When the MediaRenderer has played all of the items (i.e. all tracks have been played), then the device enters the STOPPED state and the AVT.CurrentTrack changes to "1".

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44]	393O8	

7.4.1.6.19.6

[GUIDELINE] UPnP MediaRenderers should support the "REPEAT_ONE", "REPEAT_ALL" and either "SHUFFLE" or "RANDOM" play modes.

[ATTRIBUTES]

S	L	DMR	n/a	n/a	[44]	LXYXC	

NOTE: The "NORMAL" play mode is required, but these additional play modes are only recommended.

7.4.1.6.20 MM Play Speed

7.4.1.6.20.1

[GUIDELINE] UPnP AV MediaRenderers may include element <allowedValueList> to specify a list of allowed values for AVT.TransportPlaySpeed in the service description document as defined by the AVTransport specification [44].

[ATTRIBUTES]

O	R	DMR	n/a	n/a	[44]	4WMJW	C

NOTE: This guideline indicates that listing play speed values in the service description document is optional.

Some UPnP AV MediaRenderers will only support a play speed value of '1'. These UPnP AV MediaRenderers could publish this value in the allowed-value list of the service description document.

Some UPnP AV MediaRenderers will rely on a UPnP AV MediaServer to provide support for play speed operations. These UPnP AV MediaRenderers do not know a-priori the speed values that the servers will support. These UPnP AV MediaRenderers cannot list any values in the service description document.

Some UPnP AV MediaRenderers could rely only on themselves to provide play speed support. In this case, these UPnP AV MediaRenderers could publish an exhaustive list of play speed values in the service description document.

Finally, a fourth class of UPnP AV MediaRenderers will support both server-driven and renderer-driven play speed operations. In this case, the UPnP AV MediaRenderers cannot publish an exhaustive list of speed values in the service description document and consequently, these UPnP AV MediaRenderers would also omit the allowed-value list.

If a UPnP AV MediaRenderer chooses not to specify a list of allowed play speed values, then AVT.TransportPlaySpeed will be defined in the service description document as follows:

```
<stateVariable sendEvents="no">
<name>TransportPlaySpeed</name>
<dataType>string</dataType>
</stateVariable>
```

7.4.1.6.20.2

[GUIDELINE] If a UPnP AV MediaRenderer includes the <allowedValueList> element for the AVT.TransportPlaySpeed state variable in the service description, the element shall contain all the play speed values that the UPnP AV MediaRenderer accepts as the value of the *Speed* input argument in an AVT:Play action. In addition, each value shall be represented as a rational fraction in accordance with the UPnP AVTransport specification.

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[44]	28RT5	C

NOTE: : If a DMR specifies a list of values in the service description document, then UPnP AV MediaRenderer control points know a-priori which speed values can never be used because they are unlisted. Notice that having a value in the list does not automatically indicate that the DMR will provide support for this value for a particular content resource. For any given content resource, the DMR will implement a subset of the allowed values published in the service description. The implemented values will be available to the controllers, as defined in 7.4.1.6.29.1.

Specifying allowed values like "NORMAL", "2x" or "Backwards Slow 0.25" are not allowed. Examples of correct representations are "1", "2", and "-1/4".

This guideline imposes no requirement on a control point to represent play speeds on a user interface as a rational fraction.

For example, if a UPnP AV MediaRenderer that supports speeds 1 and 4 chooses to specify a list of allowed values, then AVT.TransportPlaySpeed will be defined in the service description document as follows:

```
<stateVariable sendEvents="no">
<name>TransportPlaySpeed</name>
<dataType>string</dataType>
<allowedValueList>
     <allowedValue>1</allowedValue>
    <allowedValue>4</allowedValue>
</allowedValueList>
</stateVariable>
```

7.4.1.6.20.3

[GUIDELINE] If a UPnP AV MediaRenderer specifies a list of allowed values for AVT.TransportPlaySpeed in the service description, then each of the speed values subsequently used in the X_DLNA_PS option of AVT.CurrentTransportActions virtual instance state variable shall be one of the listed values.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44]	J7LP5

NOTE: The list of play speed values exposed by a DMR in the AVT.CurrentTransportActions virtual instance state variable has to be included in the list of allowed values exposed by the DMR in the service description document. The following examples explain better this requirement:

Example 1 (correct use)

- Allowed value list = -2, 1, 3, 5, 7, 10, 15, 20, 40
- AVT.CurrentTransportActions includes the option X_DLNA_PS=-2\,3\,40 during playback of the 1st resource.
- AVT.CurrentTransportActions includes the option X_DLNA_PS=7\,10\,15\,40 during playback of the 2nd resource.
- AVT.CurrentTransportActions includes the option X_DLNA_PS=-2\,3\,5\,7\,10\,15\,20\,40 during playback of the 3rd resource.

Example 2 (incorrect use)

- Allowed value list = -2, 1, 5, 10, 20
- AVT.CurrentTransportActions includes the option X_DLNA_PS=-8\,3\,40 during playback of the 1st resource.
- AVT.CurrentTransportActions includes the option X_DLNA_PS=1/2\,5\,20 during playback of the 2nd resource.

UPnP AV MediaRenderer control points obtain the list of speed values defined by X_DLNA_PS using the AVT:GetCurrentTransportActions action or via AVT.LastChange events.

7.4.1.6.21 MM Renderer Volume Control

7.4.1.6.21.1

[GUIDELINE] UPnP MediaRenderers that support volume control shall implement the RCS.Volume instance state variable with a range of 0 to 100, where 0 is audibly equivalent to mute and 100 is the maximum loudness. The stepping for the variable shall be 1.

A MediaRenderer implements volume control when it implements the RCS.Volume instance state variable.

[ATTRIBUTES]

M	L	DMR	n/a	n/a	[51]	3FQLR

NOTE: This guideline provides baseline expectations on how a control point can change the volume on a MediaRenderer.

The stepping of the RCS.Volume value does not have to correspond to that of actual sound volume. E.g. the actual heard volume stepping can change per five steppings of the RCS.Volume instance state variable.

7.4.1.6.21.2

[GUIDELINE] UPnP MediaRenderers that support volume control shall implement RCS:SetVolume, RCS:GetVolume, RCS:SetMute, and RCS:GetMute.

[ATTRIBUTES]

M	L	DMR	n/a	n/a	[51]	AU8BQ

7.4.1.6.22 MM DLNAQOS Support

[GUIDELINE] If DLNAQOS as defined in 7.2 is implemented, AVTransport SOAP actions shall be tagged with DLNAQOS_2, or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3), for both requests and responses in accordance with Table 11.

[ATTRIBUTES]

M	A	DMR DMC +PU+	M-DMC	MIU	n/a	VW39V

NOTE: All other forms of UPnP AV traffic described in this subclause (e.g. CDS, Rendering Service) are tagged as per the default DLNAQOS_UP value in Table 11.

7.4.1.6.23 MM Usage of AVT.CurrentTransportActions

7.4.1.6.23.1

[GUIDELINE] If a UPnP AV MediaRenderer implements controller-time, controller-byte or playspeed operations for some content types, it shall implement AVT.CurrentTransportActions virtual instance state variable, and AVT:GetCurrentTransportActions action.

"UPnP AV MediaRenderers implementing a controller-time seek operation" means: A request from a UPnP AV MediaRenderer control point to seek to some time instant "t" causes playback to re-start from time "t" (where "t" has a value larger than or equal to 0 and less than or equal to the duration of the track).

"UPnP AV MediaRenderers implementing a controller-byte seek operation" means: A request from a UPnP AV MediaRenderer control point to seek to some byte value "b" causes playback to re-start from approximately byte "b" in the stream (where "b" has a value larger than or equal to 0 and less than the track size in bytes).

"UPnP AV MediaRenderers implementing a playspeed operation" means: A request from a UPnP AV MediaRenderer control point to play at speed "s" causes playback at approximately the speed of "s", where "s" is any value other than 1.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44]	V648E

7.4.1.6.23.2

[GUIDELINE] If a UPnP AV MediaRenderer implements the AVT.CurrentTransportActions virtual instance state variable then it shall always list the available transport actions in this state variable including any of the DLNA-defined values (per guidelines 7.4.1.6.27.1 and 7.4.1.6.29.1)

[Attributes]

~~7.4~~	~~7.4~~	~~7.4.1.6.23.2.1.3~~ DMR	~~7.4.1.6.23.2.1.4~~ n/a	~~7.4.1.6.23~~ /a	[44]	M76XT

Formatted: None, No bullets or numbering, Don't keep with next, Hyphenate

NOTE: UPnP AV MediaRenderers that support controller-time seek, controller-byte seek, or playspeed operations have to implement AVT.CurrentTransportActions according to guideline 7.4.1.6.23.1. When these UPnP AV MediaRenderers play some content for which none of such operations are available, then the device still needs to show the proper list of transport actions in the state variable. For example, the available transport actions for an audio track could be: Play, Stop, Pause.

7.4.1.6.23.3

[Guideline] If no content is currently being rendered (i.e. AVT.AVTransportURI is an empty string), then a UPnP AV MediaRenderer that implements AVT.CurrentTransportActions shall use an empty string ("") as the value of the AVT.CurrentTransportActions virtual instance state variable.

[Attributes]

M	C	DMR	n/a	n/a	[44]	8RT59

7.4.1.6.24 MM DLNA State Variables for Renderer Control Operations

7.4.1.6.24.1

[Guideline] If a UPnP AV MediaRenderer implements the controller-byte seek operation for some content as defined in 7.4.1.6.23.1, then it shall include the X_DLNA_RelativeBytePosition, X_DLNA_AbsoluteBytePosition and X_DLNA_CurrentTrackSize state variables in the AVTransport service description, and shall implement the state variables.

These state variables are defined as follows:

Table 23 — DLNA state Variables for Controller-byte seek operations

Variable Name	Data Type	Allowed Value	Evented	Moderated Event
X_DLNA_RelativeBytePosition	string	Empty string (""), or a string representing an integer number in the inclusive interval: [0, (2^64) - 1]	No	No
X_DLNA_AbsoluteBytePosition	string	Empty string (""), or a string representing an integer number in the inclusive interval: [0, (2^64) - 1]	No	No
X_DLNA_CurrentTrackSize	string	Empty string (""), or a string representing an integer number in the inclusive interval: [0, (2^64)-1]	No	No

The AVT.X_DLNA_RelativeBytePosition, AVT.X_DLNA_AbsoluteBytePosition, and AVT.X_DLNA_CurrentTrackSize state variables shall not be evented via AVT.LastChange.

The X_DLNA_RelativeBytePosition state variable shall be defined in the service description document using the following XML fragment:

```
<stateVariable sendEvents="no">
   <name>X_DLNA_RelativeBytePosition</name>
   <dataType>string</dataType>
</stateVariable>
```

The X_DLNA_AbsoluteBytePosition state variable shall be defined in the service description document using the following XML fragment:

```
<stateVariable sendEvents="no">
   <name>X_DLNA_AbsoluteBytePosition</name>
   <dataType>string</dataType>
</stateVariable>
```

The X_DLNA_CurrentTrackSize state variable shall be defined in the service description document using the following XML fragment:

```
<stateVariable sendEvents="no">
   <name>X_DLNA_CurrentTrackSize</name>
   <dataType>string</dataType>
</stateVariable>
```

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44] [47]	SM76X	

NOTE: This guideline defines new state variables for the UPnP AV MediaRenderer's AVTransport service. These state variable are implemented by renderers that support controller-byte seek operations.

The first state variable, X_DLNA_RelativeBytePosition, provides the playback position in bytes during playback. Controllers that implement controller-byte seek operations can poll this state variable to determine the current byte processed by the renderer.

The second state variable, X_DLNA_AbsoluteBytePosition, is for future use. DLNA does not currently define the behavior of this state variable. Its value is vendor-dependent, as long as it conforms to the syntax defined in this guideline.

The third state variable, X_DLNA_CurrentTrackSize provides the file size information for the track currently being rendered. Controllers can poll this state variable to determine the file size.

7.4.1.6.24.2

[GUIDELINE] If a UPnP AV MediaRenderer implements the AVT.X_DLNA_RelativeBytePosition virtual instance state variable, its value shall indicate approximately the current byte in the stream processed for rendering (the current playback position measured in bytes). Byte 0 represents the first byte in the sequence. If 'L' represents the file size in bytes, then byte L-1 represents the final byte in the sequence. If no content is currently being rendered (i.e. AVT.AVTransportURI is an empty string), the value of the AVT.X_DLNA_RelativeBytePosition virtual instance state variable shall be an empty string ("").

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44] [47]	76XTD	

7.4.1.6.24.3

[GUIDELINE] If a UPnP AV MediaRenderer implements the AVT.X_DLNA_CurrentTrackSize virtual instance state variable then its value shall be one of the following:

- An integer value indicating the size (in bytes) of the track currently being rendered
- 0 to indicate that the size (in bytes) of the track currently being rendered is unknown
- An empty string ("") when no track is currently being rendered (i.e. AVT.AVTransportURI is an empty string)

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44][47]	RT59R	

7.4.1.6.24.4

[GUIDELINE] If a UPnP AV MediaRenderer does not implement the controller-byte seek operation for any type of resources, then it may omit the X_DLNA_RelativeBytePosition state variable from the AVTransport service description.

[ATTRIBUTES]

O	A	DMR	n/a	n/a	[44][47]	48EII	

7.4.1.6.24.5

[GUIDELINE] If a UPnP AV MediaRenderer does not implement the controller-byte seek operation for any type of resources, then it may omit the X_DLNA_AbsoluteBytePosition state variable from the AVTransport service description.

[ATTRIBUTES]

O	A	DMR	n/a	n/a	[44][47]	7LP5V	

7.4.1.6.24.6

[GUIDELINE] If a UPnP AV MediaRenderer does not implement the controller-byte seek operation for any type of resources, then it may omit the X_DLNA_CurrentTrackSize state variable from the AVTransport service description.

[ATTRIBUTES]

O	A	DMR	n/a	n/a	[44][47]	648EI	

7.4.1.6.25 MM DLNA actions for renderer control operations

[GUIDELINE] If a UPnP AV MediaRenderer implements the controller-byte seek operation for some content as defined in 7.4.1.6.23.1, then it shall implement the AVT:X_DLNA_GetBytePositionInfo action.

This action is defined as follows:

Table 24 — Arguments for AVT:X_DLNA_GetBytePositionInfo

Argument	Direction	relatedStateVariable
InstanceID	IN	A_ARG_TYPE_InstanceID
TrackSize	OUT	X_DLNA_CurrentTrackSize
RelByte	OUT	X_DLNA_RelativeBytePosition
AbsByte	OUT	X_DLNA_AbsoluteBytePosition

This action does not have any effect on the state. The error codes defined for this action are:

Table 25 — Error Codes for AVT:X_DLNA_GetBytePositionInfo

ErrorCode	errorDescription	Description
402	Invalid Args	Could be any of the following: not enough 'in' args, too many 'in' args, no 'in' arg by that name, one or more 'in' args are of the wrong data type.

ErrorCode	errorDescription	Description
712	Invalid InstanceID	The specified instanceID is invalid for this AVTransport.

This action shall be defined in the service description document using the following XML fragment:

```
<action>
   <name>X_DLNA_GetBytePositionInfo</name>
   <argumentList>
      <argument>
         <name>InstanceID</name>
         <direction>in</direction>
         <relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
      </argument>
      <argument>
         <name>TrackSize</name>
         <direction>out</direction>
         <relatedStateVariable>X_DLNA_CurrentTrackSize</relatedStateVariable>
      </argument>
      <argument>
         <name>RelByte</name>
         <direction>out</direction>
         <relatedStateVariable>
            X_DLNA_RelativeBytePosition
         </relatedStateVariable>
      </argument>
      <argument>
         <name>AbsByte</name>
         <direction>out</direction>
         <relatedStateVariable>
            X_DLNA_AbsoluteBytePosition
         </relatedStateVariable>
      </argument>
   </argumentList>
</action>
```

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44][47]	6XTD3	E

NOTE: This guideline defines a new action for the UPnP AV MediaRenderer AVTransport service. This action is implemented by renderers that support controller-byte seek operations. During media playback, UPnP AV MediaRenderer control points can invoke this action against the UPnP AV MediaRenderer to determine approximately the current byte position in the stream.

7.4.1.6.26 MM Seek Behavior (Control Points)

7.4.1.6.26.1

[GUIDELINE] If a UPnP AV MediaRenderer control point issues an AVT:Seek request with the *Unit* input argument equal to REL_TIME, then the value specified in the *Target* input argument shall be a time value with the same syntax and semantics defined for res@duration in guideline 7.4.1.3.8. The value shall be greater than or equal to 0.

[ATTRIBUTES]

M	A	DMC, +PU+	M-DMC	n/a	[44]	T59RR	

NOTE: In DLNA, UPnP AV MediaRenderer control points can issue AVT:Seek requests using controller-time and controller-byte variables in addition to seek operations searching for a particular track (required by [44]). This guideline specifies the range of values that can be used for the controller-time variables.

In DLNA the use of ABS_TIME is not specified. A UPnP AV MediaRenderer control point that needs to provide fast access in the case of playlists could jump between tracks (using the "seek track" mode of an AVT:Seek action), and then use controller-time, or controller-byte seek requests on the resource.

7.4.1.6.26.2

[GUIDELINE] If a UPnP AV MediaRenderer control point issues an AVT:Seek request with the *Unit* input argument equal to X_DLNA_REL_BYTE then the value specified in the *Target* input argument shall be a byte value with the same syntax and semantics defined for X_DLNA_RelativeBytePosition in guidelines 7.4.1.6.24.1 and 7.4.1.6.24.2 to indicate a byte position to seek. The value shall be greater than or equal to 0.

[ATTRIBUTES]

M	A	DMC, +PU+	M-DMC	n/a	[44]	LP5VR	

NOTE: In DLNA, UPnP AV MediaRenderer control points can issue AVT:Seek requests using controller-time, and controller-byte variables in addition to seek operations searching for a particular track (required by [44]). This guideline specifies the range of values that can be used for the controller-byte variables.

In DLNA the use of a hypothetical X_DLNA_ABS_BYTE is not specified. A UPnP AV MediaRenderer control point that needs to provide fast access in the case of media collections could jump between tracks (using the "seek track" mode of an AVT:Seek action), and then use controller-time, or controller-byte-based seek requests on the resource.

7.4.1.6.26.3

[GUIDELINE] If a UPnP AV MediaRenderer does not specify support for the controller-time operation as defined in Guideline 7.4.1.6.27.1 then UPnP AV MediaRenderer control points should not issue an AVT:Seek request with the *Unit* input argument equal to REL_TIME.

[ATTRIBUTES]

S	A	DMC, +PU+	M-DMC	n/a	[44]	GY633	

NOTE: This guideline recommends UPnP AV MediaRenderer control points to verify first if a DMR supports the controller-time seek operation before actually issuing a request.

7.4.1.6.26.4

[GUIDELINE] If a UPnP AV MediaRenderer does not specify support for the controller-byte operation as defined in Guideline 7.4.1.6.27.1 then UPnP AV MediaRenderer control points should not issue an AVT:Seek request with the *Unit* input argument equal to X_DLNA_REL_BYTE.

[ATTRIBUTES]

S	A	DMC, +PU+	M-DMC	n/a	[44]	4T7UO	

NOTE: This guideline recommends UPnP AV MediaRenderer control points to verify first if a DMR supports the controller-byte seek operation before actually issuing a request.

7.4.1.6.26.5

[GUIDELINE] If a UPnP AV MediaRenderer provides the track duration according to 7.4.1.6.27.2, then when a UPnP AV MediaRenderer control point issues an AVT:Seek request with the *Unit* input argument equal to REL_TIME, the value of the *Target* input argument should be less than the track duration.

[ATTRIBUTES]

S	A	DMC, +PU+	M-DMC	n/a	[44]	9Z83X	

7.4.1.6.26.6

[GUIDELINE] If a UPnP AV MediaRenderer provides the track size (in bytes) according to 7.4.1.6.27.5, then when a UPnP AV MediaRenderer control point issues an AVT:Seek request with the *Unit* input argument equal to X_DLNA_REL_BYTE, the value of the *Target* input argument should be less than the track size (in bytes).

[ATTRIBUTES]

S	A	DMC, +PU+	M-DMC	n/a	[44]	U48W2	

7.4.1.6.27 MM Seek Behavior (Renderers)

7.4.1.6.27.1

[GUIDELINE] If a UPnP AV MediaRenderer implements controller-time seek operations for the track currently being rendered as defined in 7.4.1.6.23.1, a UPnP AV MediaRenderer shall include "Seek" and "X_DLNA_SeekTime" in the list of comma-separated values of the AVT.CurrentTransportActions virtual instance state variable.

If a UPnP AV MediaRenderer implements controller-byte seek operations for the track currently being rendered as defined in 7.4.1.6.23.1, a UPnP AV MediaRenderer shall include "Seek" and "X_DLNA_SeekByte" in the list of comma-separated values of the AVT.CurrentTransportActions virtual instance state variable.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44]	Y633R	

NOTE: In DLNA, UPnP AV MediaRenderer control points can issue AVT:Seek requests using controller-time and controller-byte seek operations. However, a UPnP AV MediaRenderer might or might not support some of these operations for a given media resource. This guideline requires UPnP AV MediaRenderers to use specific text entries in AVT.CurrentTransportActions virtual instance state variable to indicate support for these controller seek operations. UPnP AV MediaRenderer control points check the value of this state variable using action AVT.GetCurrentTransportActions or via AVT.LastChange events to determine if a UPnP AV MediaRenderer supports the respective seek operations.

7.4.1.6.27.2

[GUIDELINE] If a UPnP AV MediaRenderer implements controller-time seek operations for the track currently being rendered as defined in 7.4.1.6.23.1, it should provide the track duration in the AVT.CurrentTrackDuration virtual instance state variable.

[ATTRIBUTES]

S	A	DMR	n/a	n/a	[44]	T7UOE	

NOTE: This guideline recommends UPnP AV MediaRenderers to provide the track duration for controller-time seek operations. Certain control points will be unable to use controller-time seek operations unless they know the playback duration.

7.4.1.6.27.3

[GUIDELINE] If a UPnP AV MediaRenderer includes the res@duration property in the <res> element that describes the resource currently being rendered, then the value of this property should be equal to the value of the AVT.CurrentTrackDuration virtual instance state variable. The <res> element that describes the resource currently being rendered is included in the AVT.CurrentTrackMetaData virtual instance state variable.

[ATTRIBUTES]

S	A	DMR	n/a	n/a	[44]	Z83XO	C

NOTE: Providing a playback duration value is a recommendation (not a requirement) for UPnP AV MediaRenderers. Although in general res@duration and AVT.CurrentTrackDuration have the same value, in some cases there could be some differences due to distinct algorithms for computing duration, etc. The res@duration property is provided by a device external to the UPnP AV MediaRenderer and AVT.CurrentTrackDuration is provided by the UPnP AV MediaRenderer.

7.4.1.6.27.4

[GUIDELINE] If a UPnP AV MediaRenderer control point detects different values for the same media resource in the AVT.CurrentTrackDuration virtual instance state variable and the res@duration property in the AVT.CurrentTrackMetaData virtual instance state variable, the UPnP control point should use the former in any interactions with the UPnP AV MediaRenderer.

[ATTRIBUTES]

S	A	DMC +PU+	M-DMC	n/a	[44]	FZL57	N

NOTE: Guideline 7.4.1.6.27.3 indicates that the values in res@duration and AVT.CurrentTrackDuration could be different because they are calculated by different devices using different algorithms. This guideline recommends UPnP control points to use the value in AVT.CurrentTrackDuration for interactions with the UPnP AV MediaRenderer; for example, sending an action to perform a controller-time seek operation.

7.4.1.6.27.5

[GUIDELINE] If a UPnP AV MediaRenderer implements controller-byte seek operations for the track currently being rendered as defined in 7.4.1.6.23.1, it should provide the track size (in bytes) in the AVT.X_DLNA_CurrentTrackSize virtual instance state variable.

[ATTRIBUTES]

S	A	DMR	n/a	n/a	[44][44]	48W2W	

7.4.1.6.27.6

[GUIDELINE] If a UPnP AV MediaRenderer includes the res@size property in the <res> element that describes the resource currently being rendered, then the value of this property should be equal to the value of the AVT.X_DLNA_CurrentTrackSize virtual instance state variable. The <res> element that describes the resource currently being rendered is included in the AVT.CurrentTrackMetaData virtual instance state variable.

[ATTRIBUTES]

C	A	DMR	n/a	n/a	[44]	633RQ	C

NOTE: Providing the resource size is a recommendation (not a requirement) for UPnP AV MediaRenderers. Although in general res@size and AVT.X_DLNA_CurrentTrackSize have the same value, in some cases there could be some differences due to content transformations, file modifications, etc.The res@size property is provided bya device external to the UPnP AV MediaRenderer and AVT.X_DLNA_CurrentTrackSize is provided bythe UPnP AV MediaRenderer.

7.4.1.6.27.7

[GUIDELINE] If a UPnP AV MediaRenderer control point detects different values for the same media resource in the AVT.X_DLNA_CurrentTrackSize virtual instance state variable and the res@size property in the AVT.CurrentTrackMetaData virtual instance state variable, the UPnP control point should use the former in any interactions with the UPnP AV MediaRenderer.

[ATTRIBUTES]

S	A	DMC +PU+	M-DMC	n/a	[44]	R6BBG	N

NOTE: Guideline 7.4.1.6.27.6 indicates that the values in res@size and AVT.X_DLNA_CurrentTrackSize could be different because of possible modifications to the media resource. This guideline recommends UPnP control points to use the value in AVT.X_DLNA_CurrentTrackSize for interactions with the UPnP AV MediaRenderer; for example, sending an action to perform a controller-byte seek operation.

7.4.1.6.27.8

[GUIDELINE] UPnP AV MediaRenderers shall not include the value X_DLNA_SeekTime in the AVT.CurrentTransportActions virtual instance state variable when rendering a track that does not belong to the Audio or AV Media Classes.

UPnP AV MediaRenderers shall not include the value X_DLNA_SeekByte in the AVT.CurrentTransportActions virtual instance state variable when rendering a track that does not belong to the Audio or AV Media Classes.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44]	7UOEU	

NOTE: Controller-time and controller-byte seek operations can only be used with audio or AV media resources. They cannot be used with other types of resources like images or XHTML-Print documents.

7.4.1.6.27.9

[GUIDELINE] If a UPnP AV MediaRenderer implements controller-byte seek operations, then it shall include X_DLNA_REL_BYTE in the allowed value list for the A_ARG_TYPE_SeekMode state variable in the AVTransport service description document.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44]	83XOG	

NOTE: DLNA defines a new type of seek mode for controller-byte seek operations. This new seek mode is triggered by using the value X_DLNA_REL_BYTE in the *Unit* argument of the AVT:Seek action. This guideline requires UPnP AV MediaRenderers that support controller-byte seek to add this value to the allowed values of AVT.A_ARG_TYPE_SeekMode.

7.4.1.6.27.10

[GUIDELINE] If a UPnP AV MediaRenderer implements controller-byte seek operations, and it receives an AVT:Seek request with a Unit value of X_DLNA_REL_BYTE and a Target value greater than or equal to the track size, then it shall respond with error code 711 (Illegal Seek Target).

[ATTRIBUTES]

M	C	DMR	n/a	n/a	[44]	8W2WO	

NOTE: Note that this guideline applies regardless of whether the UPnP AV MediaRenderer exposes the track size or not.

7.4.1.6.27.11

[GUIDELINE] If a UPnP AV MediaRenderer implements controller-time seek operations, and it receives an AVT:Seek request with a Unit value of REL_TIME and a Target value greater than the track duration, then it shall respond with error code 711 (Illegal Seek Target).

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44]	33RQQ	

NOTE: Note that this guideline applies regardless of whether the UPnP AV MediaRenderer exposes the track duration or not.

7.4.1.6.27.12

[GUIDELINE] If a UPnP AV MediaRenderer does not indicate support for controller-byte seek operations (as defined in Guideline 7.4.1.6.27.1), and if it receives an AVT:Seek request with a Unit value of X_DLNA_REL_BYTE, then the UPnP AV MediaRenderer shall respond with error code 710 (Seek Mode Not Supported).

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44]	UOEUS

NOTE: This guideline defines the error code to be used when the UPnP AV MediaRenderer Control Point requests controller-byte seek operations for the current track but the UPnP AV MediaRenderer did not advertise support for this operation.

7.4.1.6.27.13

[GUIDELINE] If a UPnP AV MediaRenderer does not indicate support for controller-time seek operations (as defined in Guideline 7.4.1.6.27.1), and if it receives an AVT:Seek request with a Unit value of REL_TIME, then the UPnP AV MediaRenderer shall respond with error code 710 (Seek Mode Not Supported).

[ATTRIBUTES]

M		DMR	n/a	n/a	[44]	3XOG3

NOTE: This guideline defines the error code to be used when the UPnP AV MediaRenderer Control Point requests controller-time seek operations for the current track but the UPnP AV MediaRenderer did not advertise support for this operation.

7.4.1.6.28 MM Play Speed Behavior (Renderers)

7.4.1.6.28.1

[GUIDELINE] A UPnP AV MediaRenderer Control Point that issues an AVT:Play action shall use a Speed argument with a value of "1" or one of the values specified by the UPnP AV MediaRenderer in the play-speed-list value (identified by X_DLNA_PS as defined in 7.4.1.6.29.2) of AVT.CurrentTransportActions virtual instance state variable.

[ATTRIBUTES]

M	A	DMC, +PU+	M-DMC	n/a	[44]	W2WOV

NOTE: UPnP AV MediaRenderer Control Points monitor AVT.CurrentTransportActions to determine the list of available play speeds for the track currently being rendered. For example, if a UPnP AV MediaRenderer exhibits the following values:

Play, Stop, Pause, Seek, X_DLNA_SeekTime, X_DLNA_PS=1/2\,4

then, the UPnP AV MediaRenderer Control Point knows that it is possible to issue an AVT:Play action with speed values of 1, ½, or 4.

7.4.1.6.29 MM Play Speed Behavior (Renderers)

7.4.1.6.29.1

[GUIDELINE] If a UPnP AV MediaRenderer implements playspeed operations for the track currently being rendered, it shall include the value "Play" and the list of available play speeds in the AVT.CurrentTransportActions virtual instance state variable in accordance with guideline 7.4.1.6.29.2.

When including the list of available play speeds into AVT.CurrentTransportActions virtual instance state variable, each comma (",") in the play speed list shall be escaped as "\,".

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44]	3RQQ8

NOTE: In DLNA, some UPnP AV MediaRenderers are capable of playback at different speeds without the help of a server (renderer-driven play speeds). Other UPnP AV MediaRenderers are capable of playback at different speeds only if the server will generate such streams (server-driven play speeds). Other UPnP AV MediaRenderers will be able to use both renderer- and server-driven play speeds.

A UPnP AV MediaRenderer informs potential controllers of its ability to operate at play speeds other than 1 for the track currently being rendered by entering an X_DLNA_PS field in the comma-separated list of AVT.CurrentTransportActions virtual instance state variable.

7.4.1.6.29.2

[GUIDELINE] A UPnP AV MediaRenderer that supports play speeds other than 1 shall include a play-speed-list value in the comma-separated list of AVT.CurrentTransportActions. The syntax and semantics for the play-speed-list value is defined as follows:

- play-speed-list="X_DLNA_PS="speed-list
- speed-list=speed*(","speed)
- speed=<conforms to the TransportPlaySpeed string, as specified in the AVTransport specification>

The value "1" shall not be included.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44]	OEUSR	

NOTE: UPnP AV MediaRenderers advertise support for play speeds by adding the list of play speeds to the available transport actions indicated in AVT.CurrentTransportActions virtual instance state variable. For example, a UPnP AV MediaRenderer that supports speeds of ½ and 4 for the current track will set AVT.CurrentTransportActions as follows:

Play, Stop, Pause, Seek, X_DLNA_SeekTime, X_DLNA_PS=1/2\,4

7.4.1.6.29.3

[GUIDELINE] UPnP AV MediaRenderers shall not include the play-speed-list value (identified by X_DLNA_PS as defined in 7.4.1.6.29.2) in the AVT.CurrentTransportActions virtual instance state variable for a track that does not belong to the Audio or AV Media Classes.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44]	XOG3H	

NOTE: The list of play speed values exposed by a UPnP AV MediaRenderer via X_DLNA_PS in AVT.CurrentTransportActions can only be used with audio or AV media resources. The list cannot be used with other types of resources like images or XHTML-Print documents.

7.4.1.6.30 MM Usage of AVT.PossiblePlaybackStorageMedia

[GUIDELINE] A UPnP AV MediaRenderer shall implement minimally the values "None" and "Network" for the AVT.PossiblePlaybackStorageMedia virtual instance state variable and the PlayMedia output parameter for the AVT:GetDeviceCapabilities action.

[ATTRIBUTES]

M	R	DMR	n/a	n/a	[44]	II9O4	N

NOTE: This is the minimal requirement to achieve the DLNA System Usages.

7.4.1.6.31 MM Mandtory Media Operations (Renderers)

7.4.1.6.31.1

[GUIDELINE] A Rendering Endpoint implementing audio or AV Media Classes shall implement all of the following media operations:

- Play (guideline 7.5.4.3.3.2 GUN:U6498)
- Stop (guideline 7.5.4.3.3.3.2 GUN:CWCV2)
- Pause (guideline 7.5.4.3.3.4.2 GUN: CZ794)
- Seek (guideline 7.5.4.3.3.7.2 GUN: PHT47))
- Pause-Release (guideline 7.5.4.3.3.5.1 GUN:3W5QP)

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[33]	S4HXE	N

7.4.1.6.31.2

[GUIDELINE] A Rendering Endpoint implementing AV Media Class shall implement all of the following media operations:

- Fast Forward Scan (guideline 7.5.4.3.3.8.2 GUN: T24LR)
- Slow Forward Scan (guideline 7.5.4.3.3.9.2 GUN: Z9BD2)
- Fast Backward Scan (guideline 7.5.4.3.3.10.2 GUN: V46LS)
- Slow Backward Scan (guideline 7.5.4.3.3.11.2 GUN: A89O5)

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[33]	XDI2P	N

NOTE: An operation might not always invoke the corresponding Media Operation due to buffering on the Rendering Endpoint.

7.4.1.6.32 MM Mandtory Media Operations (Servers)

7.4.1.6.32.1

[GUIDELINE] For every AV content binary not using DLNA Link Protection that supports “Limited Random Access Data Availability” Mode 1 or “Full Random Access Data Availability” Model (See 7.5.4.2.16 for details on mode), an HTTP Server Endpoint shall indicate support in the fourth field of the ProtocolInfo for at least on of the following:

- time-based seek
- byte-based seek

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	22FAG	N

NOTE: A content binary that is restricted to "Limited Random Access Data Availability" Mode 0 is considered live content and might have limited ability to support scan modes. (See 7.5.4.2.16.2).

7.4.1.6.32.2

[GUIDELINE] For every AV content binary not using DLNA Link Protection that supports “Limited Random Access Data Availability” Mode 1 or “Full Random Access Data Availability” Model (See 7.5.4.2.16 for details on mode), an HTTP Server Endpoint should indicate support in the fourth field of the ProtocolInfo the following:

- Playspeed

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	[33]	T8DBH	N

NOTE: A content binary that is restricted to "Limited Random Access Data Availability" Mode 0 is considered live content and might have limited ability to support scan modes. (See 7.5.4.2.16.2).

7.4.1.7 Upload & Optional Content Management Requirements

7.4.1.7.1 MM/CM: DMS with Upload Device Option Support Definition

7.4.1.7.1.1

[GUIDELINE] A UPnP AV MediaServer may support the Upload Device Option by implementing the baseline upload AnyContainer (defined in 7.4.1.7.11) and optionally the optional content management operations (OCM operations, as defined in 7.4.1.7.2).

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46] [49]	MVRMU

NOTE: This guideline means that a DMS or M-DMS can implement the Upload Device Option (i.e. can receive uploaded content from an +UP+ or M-DMU).

If the DMS or M-DMS implements the Upload Device Option, then it also implements upload AnyContainer. It can additionally support various OCM operations.

7.4.1.7.1.2

[GUIDELINE] If a UPnP AV MediaServer supports the Upload Device Option, it shall support the upload AnyContainer operation.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	WMJW5

NOTE: The baseline requirement for a DMS or M-DMS that supports the Upload Device Option is to be able to receive a CDS:CreateObject request that specifies the "DLNA.ORG_AnyContainer" as the parent container for a new CDS item. The DMS or M-DMS needs to be able to receive the content through an HTTP POST request.

The following is an example sequence of events for an upload scenario.

- The Upload Controller invokes CDS:CreateObject on the DMS. The metadata describes an image to be created in "DLNA.ORG_AnyContainer".
- The DMS approves the metadata from the CDS:CreateObject request to determine if it is valid. Since the parent container specifies "DLNA.ORG_AnyContainer", then the DMS decides that the new image object belongs in an existing CDS container (title of "New Photos") for all image uploads.
- The DMS sends the CDS:CreateObject response to the Upload Controller. The response indicates the object will be in the "New Photos" container. The response also includes a <res> element that omits a URI value but has a URI value for res@importUri.
- The Upload Controller uses an HTTP POST request to transfer the image file to the DMS.

When the DMS is able to serve the new image, it provides a URI value for the <res> element.

7.4.1.7.2 MM/CM: Optional Content Management Operation Definitions

[GUIDELINE] If a UPnP AV MediaServer supports optional content management (OCM) operations, it may support one or more of the following OCM operations.

- OCM: upload content Use this to upload content to a specific CDS container. This operation has 2 steps: use CDS:CreateObject to create a CDS item and use HTTP POST to transfer the content.
- OCM: create child container Use this to create a new CDS container in a specified CDS container. This operation has one step: use CDS:CreateObject to create a CDS container. This operation used generally as a preceding step to an OCM: upload content operation.
- OCM: destroy object Use this to destroy a CDS object. This operation has one step: use CDS:DestroyObject to destroy a CDS object
- OCM:change metadata Use this to alter the metadata of an existing CDS item. This operations has one step: use CDS:UpdateObject to update the CDS metadata. Note that

this operation can be used to add, delete, or change an existing metadata element of an item.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46] [49]	FQLR9	C

NOTE: The guidelines define interoperability specifically for the described usages. Vendors need to always implement behavior that is consistent with the DLNA guidelines, even if the implementation will be used for an operation that has different preconditions. For example, a DMS that supports the OCM: create child container can allow control points to create CDS containers. In such a scenario, the DMS and control point need to abide by appropriate syntax rules for CDS:CreateObject.

The guideline also permits other forms of management operations, although the guidelines do not define interoperability rules for them.

Note: OCM: destroy item has been replaced by OCM: destroy object. All original functionality on items is retained; however, revision to guidelines now allows Destroy operations on containers.

7.4.1.7.3 MM/CM: Upload Controller and Mobile Digital Media Uploader

7.4.1.7.3.1

[GUIDELINE] A DLNA device class may implement the Upload Controller Device Capability.

[ATTRIBUTES]

O	A	DMS DMP DMR DMC DMPr	M-DMS M-DMP M-DMC M-DMD	n/a	[46] [49]	U8BQB	

7.4.1.7.3.2

[GUIDELINE] An Upload Controller or a Mobile Digital Media Uploader shall implement a UPnP AV MediaServer control point capable of invoking the following actions.

- CDS:CreateObject.

[ATTRIBUTES]

M	C	+UP+	M-DMU	n/a	[46] [49]	W39VI	

NOTE: CDS:CreateObject is the action that allows an Upload Controller or M-DMU to upload content. The guidelines specify that the normative content transfer methodology involves HTTP POST, as described in guideline 7.5.4.3.6.1.1.

7.4.1.7.3.3

[GUIDELINE] An Upload Controller or M-DMU shall support the upload AnyContainer operation.

[ATTRIBUTES]

M	C	+UP+	M-DMU	n/a	[46] [49]	VRMUZ	

7.4.1.7.3.4

[GUIDELINE] An Upload Controller or M-DMU may implement a UPnP AV MediaServer control point capable of invoking the following actions.

- CDS:DestroyObject

By supporting these actions, an Upload Controller can be capable of supporting additional optional content management operations.

See 7.4.1.7.2 for more information about optional content management operations.

[ATTRIBUTES]

O	C	+UP+	M-DMU	n/a	[46] [49]	MJW54

NOTE: CDS:DestroyObject is the action that allows an Upload Controller to destroy a CDS object when a content transfer failed or when the user wants to remove uploaded content from the DMS.

Upload Controllers and M-DMUs are allowed to implement support for other CDS actions, but the DLNA guidelines do not specify the interoperability behavior for other actions.

7.4.1.7.4 MM/CM: Determining Upload AnyContainer Support

7.4.1.7.4.1

[GUIDELINE] A UPnP MediaServer shall use the <dlna:X_DLNACAP> element (as a child of the <device> element that represents the MediaServer) in the device description document and use the following Capability IDs in the element's comma-separated value list to indicate support for uploading a media class.

Table 26 — Capability IDs for AnyContainer Support

Capability ID	Description
audio-upload	The UPnP AV MediaServer supports the upload AnyContainer operation for the Audio media class.
image-upload	The UPnP AV MediaServer supports the upload AnyContainer operation for the image media class.
av-upload	The UPnP AV MediaServer supports the upload AnyContainer operation for the AV media class.
create-child-container	The UPnP AV MediaServer supports the OCM: create child container opperation.
create-item-with-OCM-destroy-item	The UPnP AV MediaServer supports to create CDS item with OCM: destroy object capability for the upload AnyContainer operation. This Capability ID shall coexist with at least one of audio-upload, image-upload or av-upload.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	QLR9B

NOTE: DMS devices use the <dlna:X_DLNACAP> element to indicate support for the upload AnyContainer operation. The element is a comma separated value list that indicates whether the DMS can receive uploads of images, audio-only, or audio/video content.

Note that a DMS that supports the upload AnyContainer operation is different than a DMS with an Upload Controller. The former is a DMS that can receive uploaded content. The latter is a DMS that can upload to a different DMS. It is possible to implement a DMS that supports the upload AnyContainer operation and the Upload Controller capability.

A DMS that supports OCM: create child container need to also support the creation of child containers where the DMS chooses the parent container (because the Upload Controller specified DLNA.ORG_AnyContainer as the parent). This guideline explains how Upload Controllers determine if OCM: create child container is supported for the DMS.

See guideline 7.3.2.35.1 for the formal syntax of the <dlna:X_DLNACAP> element.

7.4.1.7.4.2

[GUIDELINE] A UPnP AV MediaServer may implement the CDS:X_GetDLNAUploadProfiles action to indicate the DLNA media format profiles that it will accept in the CDS:CreateObject action.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46] [49]	8BQBA

7.4.1.7.4.3

[GUIDELINE] The CDS:X_GetDLNAUploadProfiles action's definition in the service description shall be defined as indicated below.

```
<action>
   <name>X_GetDLNAUploadProfiles</name>
   <argumentList>
      <argument>
         <name>UploadProfiles</name>
         <direction>in</direction>
         <relatedStateVariable>
            X_A_ARG_Type_UploadProfiles
         </relatedStateVariable>
      </argument>
      <argument>
         <name>SupportedUploadProfiles</name>
         <direction>out</direction>
         <relatedStateVariable>
            X_A_ARG_Type_SupportedUploadProfiles
         </relatedStateVariable>
      </argument>
   </argumentList>
</action>
```

The X_A_ARG_TYPE_UploadProfiles and X_A_ARG_Type_SupportedUploadProfiles state variables are defined below.

```
<stateVariable sendEvents="no">
   <name>X_A_ARG_Type_UploadProfiles</name>
   <dataType>string</dataType>
</stateVariable>
<stateVariable sendEvents="no">
   <name>X_A_ARG_Type_SupportedUploadProfiles</name>
   <dataType>string</dataType>
</stateVariable>
```

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	39VIY

NOTE: The *UploadProfiles* input argument is an unordered, comma separated list of DLNA media format profile names.

The *SupportedUploadProfiles* output argument is an unordered, comma separated list of DLNA media format profile names, as described below.

- Are listed in the *UploadProfiles* input argument and this MediaServer is willing to accept at the action is invoked.
- Or, in case of empty *UploadProfiles* input argument, the SupportedUploadProfiles list will contain the complete list of DLNA media format profiles that this MediaServer is willing to accept at the time the action is invoked.

The DLNA media profile IDs that appear in SupportedUploadProfiles needs to comply with these restrictions.

- Needs to be AV, Audio, or Image media classes.
- Media format profile IDs for icons, thumbnails, media collection files, and XHTML print documents are expressly prohibited.

The response behavior is summarized in the following way.

- If UploadProfiles is empty, then SupportedUploadProfiles contains a complete list of profiles that the MediaServer is willing to accept at the current time. Control points specify an empty value for UploadProfiles when they want to get a full list of profiles that the MediaServer will accept for uploads.

- If UploadProfiles contains one or more profiles, then SupportedUploadProfiles contains the subset of UploadProfiles that the MediaServer is willing to accept at the current time. Control points specify one or more profiles for UploadProfiles when they are interested in uploading specific formats to a MediaServer.

7.4.1.7.4.4

[Guideline] If a UPnP AV MediaServer does not accept upload of all the DLNA media format profiles that it lists in the CMS.SourceProtocolInfo state variable, then it shall implement the CDS:X_GetDLNAUploadProfiles action to indicate the DLNA media format profiles that it will accept in the CDS:CreateObject action.

[Attributes]

M	A	DMS	M-DMS	n/a	[46] [49]	N7IEV	E

NOTE: A UPnP AV MediaServer does not need to implement the optional CDS:X_GetDLNAUploadProfiles action if it supports the same set of DLNA media format profiles for upload and for content serving.

A UPnP AV MediaServer control point needs to realize that the presence of this action is the first indicator that there are restrictions in the uploadable media format profiles.

7.4.1.7.4.5

[Guideline] If a UPnP MediaServer supports the upload AnyContainer operation or OCM:create child container, then it shall adhere to the following guidelines (7.4.1.7.4 through 7.4.1.7.4.4).

[Attributes]

M	A	DMS	M-DMS	n/a	[46] [49]	4R8S6	

7.4.1.7.5 MM/CM: Operations that need CDS:CreateObject

[Guideline] If a UPnP AV MediaServer supports one or more of these operations, then it shall implement CDS:CreateObject.

- upload AnyContainer operation
- OCM: upload content
- OCM: create child container

[Attributes]

M	A	DMS	M-DMS	n/a	[46] [49]	RMUZ5	

The CDS:CreateObject action is used to create a CDS object that will represent the uploaded content.

In addition to these guidelines, a DMS or M-DMS with the ability to receive uploaded content needs to implement an HTTP server capable of processing HTTP POST requests, as described in 7.5.4.3.6.1 guidelines.

7.4.1.7.6 MM/CM: Operations that need CDS:DestroyObject

[Guideline] If a UPnP AV MediaServer supports this operation, then it shall implement CDS:DestroyObject.

- OCM: destroy object

[Attributes]

M	A	DMS	M-DMS	n/a	[46] [49]	MUZ56	

NOTE: The CDS:DestroyObject action is used for a variety of OCM operations related to removing CDS objects from a DMS.

7.4.1.7.7 MM/CM: Other CDS Actions

[GUIDELINE] A UPnP AV MediaServer may implement CDS:DeleteResource, CDS:CreateReference, CDS:ImportResource, CDS:ExportResource, or CDS:StopTransferResource.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46] [49]	9VIYY	

NOTE: These are normative UPnP AV CDS actions, but the DLNA guidelines do not define interoperability rules for them.

7.4.1.7.8 MM/CM: Baseline Media Formats

7.4.1.7.8.1

[GUIDELINE] A UPnP AV MediaServer that belongs to the HND Device Category and implements the upload AnyContainer operation for a DLNA Media Class (as indicated by guideline 7.4.1.7.4.1) shall accept content uploads of the mandatory Media Format Profiles that the UPnP AV MediaServer is capable of exposing according to [56], guideline 6.1.1.2, of for that DLNA Media Class in the HND Device Category.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46] [49] [56]	BQBAR	

NOTE: DLNA guidelines for HND Device Classes have an expressed goal to facilitate uploading of baseline media formats for exposing and rendering content. A DMS that only supports uploads of optional media formats detracts from the guidelines' interoperability message.

7.4.1.7.8.2

[GUIDELINE] A UPnP AV MediaServer that belongs to the MHD Device Category and implements the upload AnyContainer operation for a DLNA Media Class (as indicated by guideline 7.4.1.7.4.1) shall accept the uploading of content uploads of all the mandatory Media Format Profiles for that DLNA Media Class in the MHD Device Category..

[ATTRIBUTES]

M	A	n/a	M-DMS	n/a	[46] [49]	SR7JJ	C

NOTE: DLNA guidelines for MHD Device Classes have an expressed goal to facilitate uploading of baseline media formats for exchanging content and exposing and rendering content. An M-DMS that only supports uploads of optional media formats detracts from the guidelines' interoperability message.

7.4.1.7.8.3

[GUIDELINE] A UPnP AV MediaServer control point that belongs to one of the DLNA-defined Device Categories and implements the upload AnyContainer operation for a DLNA Media Class shall be able to upload content items of at least one of the mandatory DLNA Media Format Profiles for that DLNA Media Class in its Device Category.

Being able to upload a content item of a DLNA Media Format Profile means that, given a content item of that DLNA Media Format Profile, the UPnP AV MediaServer control point shall be able to issue the CDS:CreateObject request with the correct DLNA.ORG_PN parameter in the fourth field of ProtocolInfo value of the <res> element to represent the DLNA Media Format Profile of the content item. (See 7.4.1.3.17).

[ATTRIBUTES]

M	A	+UP+	D_DMU	n/a	[46]	ZRERW	C

7.4.1.7.8.4

[GUIDELINE] A UPnP AV MediaServer that belongs to the DLNA-defined DMS Device Class and implements the Content Synchronization Device Option (as indicated by 7.4.3.5.2) shall support the upload of the mandatory DLNA Media Format Profiles in both the HND and MHD Device Categories for the supported DLNA Media Classes.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46] [49]	XLZ4X	E

NOTE: Facilitates the content synchronization process between MHD devices and HND devices (e.g. DMS). The guideline is not applicable to M-DMS devices.

See [56] 6.2.

7.4.1.7.8.5

[GUIDELINE] A UPnP AV MediaServer that belongs to the DLNA-defined M-DMS Device Class and implements the Content Synchronization Device Option (as indicated by 7.4.3.5.2) shall support the upload of the mandatory DLNA Media Format Profiles in only the MHD Device Categories for the supported DLNA Media Classes.

[ATTRIBUTES]

M	A	n/a	M-DMS	n/a	[46] [49]	8CGDS	E

NOTE: MHD devices are only expected to Synchronize with other MHD devices.

See [56] 6.2.

7.4.1.7.9 MM/CM: Indicating Support for OCM Operations

7.4.1.7.9.1

[GUIDELINE] If a UPnP AV MediaServer supports one or more OCM operations, then the UPnP AV MediaServer may have one or more CDS objects with the @dlna:dlnaManaged attribute.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46] [49]	LR9BX	

NOTE: The @dlna:dlnaManaged attribute indicates the OCM operations that the DMS or M-DMS is able to support for a given CDS object.

7.4.1.7.9.2

[GUIDELINE] If a UPnP AV MediaServer supports one or more OCM operations on a CDS object, then the UPnP AV MediaServer shall use the @dlna:dlnaManaged attribute to indicate support for those OCM operations on a CDS object.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	JW54B	

7.4.1.7.9.3

[GUIDELINE] If a CDS object allows one or more OCM operations, then the CDS object shall have a @restricted value of "0".

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	XYXC6	

NOTE: For CDS objects that do not allow OCM operations, the @restricted attribute can have a value of "0" or "1" to indicate that the object allows or disallows modifications through non-OCM operations respectively.

7.4.1.7.9.4

[Guideline] The syntax definition for the value of @dlna:dlnaManaged attribute shall be as follows:

- dlnaManaged-value = 8 hexdigit
- hexdigit = <hexadecimal digit: "0"-"9", "A"-"F", "a"-"f">

The @dlna:dlnaManaged attribute is a 32-bit unsigned integer encoded into exactly 8 hexadecimal digits, with the following bit definitions. Bit-0 is the least significant bit. If a bit supports a particular operation, then the bit value is true. Otherwise, the bit value is false to indicate the operation is not supported. (e.g. 00000000000000000000000000000001b = 0x00000001 where bit-0 is set to true).

Example:

- dlna:dlnaManaged="00000001"

The hexadecimal encoded form shall consist only of hexadecimal digits. The value shall omit the "0x" string that often precedes hexadecimal notation.

- Bit-0: indicates support for OCM: upload content
  - If true then the MediaServer allows a control point to create child CDS items in the container for the OCM: upload content operation.
  - Shall be false when used with a CDS item.
- Bit-1: indicates support for OCM: create child container
  - If true on a CDS container, then the MediaServer allows a control point to create child CDS containers that can support the OCM: upload content.
  - Shall be false when used with a CDS item.
- Bit-2: indicates support for OCM: destroy object operation
  - If true then the MediaServer allows a control point to perform an OCM: destroy object operation on the object.
- Bit-3: indicates support for OCM: upload content with OCM:destroy object operation capability
  - If true on a CDS container, then the MediaServer allows a control point to create CDS items for OCM:upload content operation that can support the OCM: destroy object.
  - If true on a CDS container, then Bit-0 value on the CDS container shall be true.
  - Shall be false when used with a CDS item.
- Bit 4: indicates support for OCM: change metadata operation
  - If true then the MediaServer allows a control point to change, add or delete metadata on an existing CDS Object.
- All other bits shall be false. All other bits are reserved for future use.

[Attributes]

M	A	DMS	M-DMS	n/a	[46] [49]	93O8Z	C

NOTE: This guideline defines the syntax for the @dlna:dlnaManaged value. This attribute describes the DMS implementation's ability to support operations that affect the CDS object as a whole.

Note that a false value for a bit means that the DMS or M-DMS does not claim support for the described operation. Control points are expected to honor the interpretation of these bits to maximize interoperability because invoking a CDS action that is related to an unsupported OCM operation are likely to receive an error. CDS failure responses are not mandatory when a DMS or M-DMS detects a deviation because vendors are permitted to use normative CDS actions for vendor-defined operation.

There can be cases where a DMS (that is normally able to destroy the CDS item) is not able to destroy a CDS item at the time of the request. For example, the actual content binary file might be locked or a local management policy prevents the CDS item from being destroyed at the current moment."

7.4.1.7.9.5

[GUIDELINE] If the @restricted attribute is set to '1' all bits of the @dlna:dlnaManaged attribute shall be false - no OCM operations are allowed.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	B73OL	

NOTE: Enforce consistency between the @restricted and the @dlna:dlnaManaged attributes.

7.4.1.7.9.6

[GUIDELINE] If a CDS object has “0” value for all hexadecimal digits in the @dlna:dlnaManaged attribute, then UPnP AV MediaServer should omit the @dlna:dlnaManaged attribute of the CDS object.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46] [49]	YHXEF	N

NOTE: This guideline provides guidance to UPnP AV MediaServer implementations to not assign the @dlna:dlnaManaged=“00000000” attribute to a CDS object.

7.4.1.7.10 MM/CM: Parallel Upload AnyContainer and OCM operations

7.4.1.7.10.1

[GUIDELINE] If a MediaServer control point attempts to do multiple upload AnyContainer or multiple OCM operations in parallel, and if the MediaServer fails one or more of the parallel attempts, then the MediaServer control point shall be able to perform the upload operations in a serialized manner.

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	6X6HM	

NOTE: DMS devices are not required to support parallel attempts to upload or destroy content. Control points that attempt to do so are responsible for retrying in a serialized manner in the event of a failure. The exact user's process for retrying serialized uploads is a user interface issue and is out of scope.

7.4.1.7.10.2

[GUIDELINE] If a UPnP AV MediaServer supports the upload AnyContainer or other OCM operations, then it shall be capable of performing at least one upload AnyContainer or OCM operation at a time.

[ATTRIBUTES]

M	A	DMS	M-DMS	MIU	[46] [49]	N9C8F	

7.4.1.7.11 MM/CM: Upload AnyContainer Operation

7.4.1.7.11.1

[GUIDELINE] If a UPnP AV MediaServer supports the upload AnyContainer operation, then it shall allow control points to specify a "DLNA.ORG_AnyContainer" value for the *ContainerID* input argument in a CDS:CreateObject request.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	4ZYSH

NOTE: The "DLNA.ORG_AnyContainer" allows an Upload Controller to upload content to a UPnP AV MediaServer without having specific knowledge of where the content will be listed in the CDS hierarchy.

Where the MediaServer creates the new object is dependent on the MediaServer implementation.

The "DLNA.ORG_AnyContainer" is a magic container ID which is used only in the request and a container of this container ID does actually not exist.

Note that a control point can attempt to deviate slightly from the restrictions listed in 7.4.1.7.11, but the result can be an error. For example, if the Upload Controller tries to use object.item.audioItem.audioBroadcast instead of object.item.audioItem, then the DMS can choose to fail the request.

7.4.1.7.11.2

[GUIDELINE] If a UPnP AV MediaServer supports the upload AnyContainer operation, then the @id value for each container shall be a value that is not equal to "DLNA.ORG_AnyContainer".

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	LSFB4

7.4.1.7.11.3

[GUIDELINE] If a UPnP AV MediaServer control point is going to start an upload AnyContainer operation, then it shall invoke CDS:CreateObject with the following rules.

- The *ContainerID* input argument shall be "DLNA.ORG_AnyContainer".
- The *Elements* input argument shall specify a CDS item with a single <res> element that does not have a URI value. The <res> element shall also conform to guideline 7.4.1.7.19. The <item> element shall also have a <upnp:class> value (or similarly derived value) that corresponds to the media class of the content that is going to be uploaded.

Table 27 — Required Media Class UPnP Values

Media Class	Required UPnP:class value
Audio	object.item.audioItem
Image	object.item.imageItem
AV	object.item.videoItem

[ATTRIBUTES]

M	A	+UP+	M-DMU	MIU	[46] [49]	SFB47

7.4.1.7.11.4

[GUIDELINE] If a UPnP AV MediaServer receives an upload AnyContainer request with a <upnp:class> value that is derived from a supported base class value, then the DMS may change the <upnp:class> value to the supported base class or to a similarly derived class, as indicated in the *Result* output argument of the CDS:CreateObject.

For example, if the request specifies object.item.audioItem.audioBroadcast, then the MediaServer can change the value to object.item.audioItem. Likewise, if the request specified object.item.imageItem, the MediaServer can change it to object.item.imageItem.photo.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[46] [49]	ZYSHA

NOTE: DMS devices are not required to support the full range of derived values for the <upnp:class> element.

7.4.1.7.11.5

[GUIDELINE] If a UPnP AV MediaServer returns a success response to an Upload Controller's request to start an upload AnyContainer operation, then the MediaServer shall do the following.

- The DMS or M-DMS device shall determine an appropriate CDS container where the new CDS object will be created.
- The MediaServer shall specify the object ID of the parent container as the <item> element's @parentID attribute value, which is returned in the *Result* output argument.
- The <res> element (found in the *Result* output argument) that provides the res@importUri value intended for the content transfer process shall comply with the guidelines in 7.4.1.7.19.2.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	9C8FT

NOTE: This guideline describes the proper DMS or M-DMS behavior in a success scenario. The DMS needs to use an appropriate DMS error in the case of a failure.

7.4.1.7.11.6

[GUIDELINE] A UPnP AV MediaServer that supports the Upload AnyContainer operation shall be capable of participating in a content transfer process, as described in guideline 7.4.1.7.26.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	X6HMR

NOTE: These guidelines introduce the second step to the Upload AnyContainer operation.

7.4.1.7.11.7

[GUIDELINE] A UPnP AV MediaServer control point that supports the Upload AnyContainer operation shall be capable of participating in a content transfer process, as described in guidelines 7.4.1.7.26.

[ATTRIBUTES]

M	A	+UP+	M-DMU	MIU	[46] [49]	RGKUS

7.4.1.7.11.8

[GUIDELINE] A UPnP AV MediaServer that creates a new CDS item in an upload AnyContainer operation may create the CDS item in a CDS container with the @dlna:dlnaManaged attribute.

[Attributes]

O	C	DMS	M-DMS	n/a	[46] [49]	3O8ZS	

NOTE: DMS or M-DMS devices can create the new CDS item in a CDS container that has the @dlna:dlnaManaged attribute.

7.4.1.7.11.9

[Guideline] A UPnP AV MediaServer control point shall tolerate scenarios where the MediaServer changes values to metadata properties specified in a CDS:CreateObject request.

Tolerate means that the UPnP AV MediaServer control point is able to complete the necessary content transfer process (including the transfer of IFO files, if necessary) after creating the CDS object.

[Attributes]

M	C	+UP+	M-DMU	MIU	[46] [49]	YXC6T	

NOTE: The following are some examples of what a MediaServer is permitted to do.

- The @parentID changes from "DLNA.ORG_AnyContainer" to a different @parentID value.
- The <upnp:class> value changes to a derived class or to a base class of the current value.
- <dc:title>, <dc:creator>, and other string-based user-informational metadata properties are truncated to fit the maximum length supported by the MediaServer.

7.4.1.7.11.10

[Guideline] A UPnP AV MediaServer that creates a new CDS item in an upload AnyContainer operation may change metadata values to indicate the support or unsupported nature of OCM operations.

[Attributes]

O	C	DMS	M-DMS	n/a	[46] [49]	W54BN	

NOTE: MediaServers are permitted to change metadata values at any time. This behavior is also permissible when the MediaServer actually creates the new CDS object.

7.4.1.7.12 MM/CM OCM: Upload Content Operation

7.4.1.7.12.1

[Guideline] If a UPnP AV MediaServer supports the OCM: upload content operation on a CDS container, then it shall specify one or more <upnp:createClass> elements to indicate the types of CDS objects that can be created in the container. The values of these upnp:createClass elements shall be equal to or derived from object.item.

[Attributes]

M	R	DMS	M-DMS	n/a	[46] [49]	R9BXN	

NOTE: This guideline ensures that control points will know what type of CDS objects can be created in the container.

For example, if a container supports the creation of image and audio-only objects, then it would use a <upnp:createClass> element with object.item.imageItem and a <upnp:clreateClass> element with object.item.audioItem.

Please note that the CDS container can have <upnp:createClass> that are derived from object.container, as described in 7.4.1.7.13.

7.4.1.7.12.2

[GUIDELINE] If a UPnP AV MediaServer control point is going to start an OCM: upload content operation, then it shall invoke CDS:CreateObject with the following rules.

- The *ContainerID* input argument shall indicate a CDS container that supports the OCM: upload content operation.
- The *Elements* input argument shall be a DIDL-Lite XML fragment that has a CDS item whose <upnp:class> value matches one of the values in the set of <upnp:createClass> elements associated with the CDS container identified by the *ContainerID* input argument. If the upnp:createClass@includeDerived attribute has a value of "1", then the <upnp:class> value also matches against the classes derived from the <upnp:createClass> value. The <item> element shall also have a single <res> element that does not specify a URI value. The <res> element shall also conform with guidelines in 7.4.1.7.19.

[ATTRIBUTES]

M	L	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	QBARR

NOTE: If a MediaServer supports the operation, Upload Controllers and M-DMUs can choose to upload content to specific CDS containers using the OCM: upload content operation.

Note that a control point can attempt to deviate slightly from the restrictions listed in 7.4.1.7.12.2, but the result can be an error. For example, if the Upload Controller tries to specify multiple <res> elements, then the DMS can choose to fail the request.

7.4.1.7.12.3

[GUIDELINE] If a UPnP AV MediaServer responds with a success to a CDS:CreateObject request for an OCM: upload content operation, then the created CDS item (as returned in the *Result* output argument) shall comply with the following rules.

- The @parentID of the created CDS item shall match the request's specified *ContainerID* input argument.
- The <res> element that provides the res@importUri value intended for the content transfer process shall comply with the guidelines in 7.4.1.7.19.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	VIYY4

7.4.1.7.12.4

[GUIDELINE] A UPnP AV MediaServer that supports the OCM: upload content operation shall be capable of participating in a content transfer process, as described in guideline 7.4.1.7.26.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	UZ56H

NOTE: These guidelines introduce the second step to the OCM: content upload operation.

7.4.1.7.12.5

[GUIDELINE] A UPnP AV MediaServer control point that supports the OCM: content upload operation shall be capable of participating in a content transfer process, as described in guideline 7.4.1.7.26.

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	Z56HE

7.4.1.7.12.6

[Guideline] A UPnP AV MediaServer that creates a new CDS item in an OCM: upload content operation may change metadata values to indicate the support or unsupported nature of OCM operations.

[Attributes]

O	C	DMS	M-DMS	n/a	[46] [49]	IYY4H	

NOTE: For example, the DMS can automatically create the @dlna:dlnaManaged attribute or change the @restricted value to indicate support for one or more OCM operations.

7.4.1.7.13 MM/CM: OCM: Create Child Container Operation

7.4.1.7.13.1

[Guideline] If a UPnP AV MediaServer supports the OCM: create child container operation on a CDS container, then it shall specify one or more <upnp:createClass> elements to indicate the types of CDS objects that can be created in the container.

[Attributes]

M	R	DMS	M-DMS	n/a	[46] [49]	YY4H4	

NOTE: This guideline ensures that control points will know what type of CDS containers can be created in an existing container.

Please note that destroying containers is out-of-scope for this version of DLNA guidelines. DLNA assumes that DMS have ownership of their CDS hierarchy, which includes the ability to remove containers through out-of-band mechanisms.

7.4.1.7.13.2

[Guideline] If a UPnP AV MediaServer supports the OCM: create child container operation, then it shall allow control points to specify a "DLNA.ORG_AnyContainer" value for the *ContainerID* input argument in a CDS:CreateObject request.

[Attributes]

M	A	DMS	M-DMS	n/a	[46] [49]	56HEF	

NOTE: The "DLNA.ORG_AnyContainer" allows an Upload Controller to create a container on a UPnP AV MediaServer without having specific knowledge of where the content will be listed in the CDS hierarchy.

7.4.1.7.13.3

[Guideline] A UPnP AV MediaServer that supports OCM: create child container shall also implement support for OCM: upload content.

[Attributes]

M	A	DMS	M-DMS	n/a	n/a	BXNX4	

7.4.1.7.13.4

[Guideline] If a UPnP AV MediaServer control point is going to start an OCM: create child container operation, then it shall invoke CDS:CreateObject with the following rules.

- The *ContainerID* input argument shall indicate "DLNA.ORG_AnyContainer" or a CDS container that supports the OCM: create child container operation.
- The *Elements* input argument shall be a CDS container whose <upnp:class> value matches one of the values in the set of <upnp:createClass> elements associated with the CDS container identified by the *ContainerID* input argument. If specifying

"DLNA.ORG_AnyContainer" as the value for the *ContainerID* input argument, then the value shall be object.container or a similarly derived class.

- The *Elements* input argument shall include one or more <upnp:createClass> values that describe the types of CDS objects that will be created in the new container. If specifying "DLNA.ORG_AnyContainer" as the value for the *ContainerID* input argument, then the corresponding <upnp:createClass> (or similarly derived values) shall be used.

Table 28 — Required UPnP createClass Elements

Media Class	Required upnp:class value (for upnp:createClass elements)
Audio	object.item.audioItem
Image	object.item.imageItem
AV	object.item.videoItem

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	9BXNX

NOTE: If a MediaServer supports the operation, Upload Controllers or M-DMUs can choose to create new CDS containers that can receive uploaded content in an organized way.

The <upnp:createClass> values that are declared in the CDS:CreateObject request indicate the types of media that the Upload Controller intends to upload into the new container.

Note that the following conditions can cause the DMS to return an error because specific aspects of the syntax are optional.

- The <upnp:class> value is derived from object.container. DMS can fail this request because object.container is the only value that is mandatory.
- The <upnp:createClass> value is unsupported, even if it is derived from a supported type. DMS can fail this request because only the upnp:createClass values listed in the table are mandatory.
- There is more than one <upnp:createClass> value in the request. DMS can fail this request because only a single <upnp:createClass> is required.

7.4.1.7.13.5

[GUIDELINE] A UPnP AV MediaServer may change a <upnp:createClass> or <upnp:class> value to one of the supported base classes or one of the derived classes if the value is derived from that base class.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46] [49]	54BN6

NOTE: For example, if the Upload Controller specified a <upnp:createClass> value of object.item.imageItem.photo and a <upnp:class> value of object.container.album, then the DMS can automatically change the values of those elements to object.item.imageItem and object.container, respectively. Similarly, a DMS can change object.container to a derived value, such as object.container.storageFolder.

Creating a container that has <upnp:createClass> values derived from object.container are out of scope.

Guideline 7.4.1.7.25.3 instructs a DMS to return error code 712 if the specified value is not acceptable to the DMS.

7.4.1.7.13.6

[GUIDELINE] A UPnP AV MediaServer control point creates a new CDS container with intent to follow up with an OCM: upload content operation the new container, then the control point shall specify the @dlna:dlnaManaged attribute with bit-0 set to true.

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	4BN6R	

NOTE: In conjunction with the <upnp:createClass> elements, this guideline ensures that a DMS knows the intent of the Upload Controller to upload content to the new container.

7.4.1.7.13.7

[GUIDELINE] A UPnP AV MediaServer that creates a new CDS container in an OCM: create child container operation may change the @restricted and/or @dlna:dlnaManaged metadata values to indicate the support or unsupported nature of OCM operations.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[46] [49]	XC6TY	

7.4.1.7.13.8

[GUIDELINE] If a UPnP AV MediaServer receives an OCM: create child container request and one or more of the <upnp:createClass> values specified in the request are unsupported, then the MediaServer shall return error code 712 (Bad Metadata).

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46] [49]	C6TY8	

7.4.1.7.13.9

[GUIDELINE] If a UPnP AV MediaServer responds with a success to a CDS:CreateObject request for an OCM: create child container operation, then the created CDS container shall have the @dlna:dlnaManaged attribute. Furthermore, the @parentID of the created CDS container (as returned in the *Result* output argument) shall match the request's specified *ContainerID* input argument.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	O8ZSQ	

NOTE: The @dlna:dlnaManaged attribute is always present in a recursive manner. However, the individual bits on the attribute value can have different true/false values.

Note that guideline 7.4.1.7.9.4 requires the created container to support the OCM:upload contents operation.

7.4.1.7.14 MM/CM: OCM: Destroy Object Operation

7.4.1.7.14.1

[GUIDELINE] If a UPnP AV MediaServer control point is going to start an OCM: destroy object operation, then it shall invoke CDS:DestroyObject with the following rules.

- The *ObjectID* input argument shall indicate a CDS object that supports the OCM: destroy object operation.
- This *ObjectID* shall be for a CDS object that is to be removed from the CDS.

[ATTRIBUTES]

M	R	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	8ZSQB	

NOTE: The primary usage of this operation is to allow an M-DMU Upload Controller or Synchronization Controller to remove CDS objects from a MediaServer. A control point can make no assumptions about whether any content files will actually be removed.

7.4.1.7.14.2

[GUIDELINE] If a UPnP AV MediaServer responds with a success to a CDS:DestroyObject request for an OCM: destroy object operation, then the DMS shall remove the CDS item indicated by the request's *ObjectID* input argument. A MediaServer that cannot remove the indicated CDS item from the CDS hierarchy shall return a SOAP error response.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46] [49]	6HMRW

NOTE: DLNA does not specify any mandatory behavior for whether or not actual content binaries are removed from the local storage of the DMS. The primary expectation of this guideline is that the destroyed CDS item no longer appears in the CDS hierarchy.

7.4.1.7.14.3

[GUIDELINE] If a UPnP AV MediaServer supports the OCM:destroy object Operation and the control point invokes CDS:DestroyObject on a container where the container and all of its descendent objects have bit 2 of the @dlna.dlnaManaged attribute true, then the container and all of its descendent objects are removed.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46] [49]	GAMUR

NOTE: Bit 2 of the @dlna.dlnaManaged attribute is the bit that defines whether the OCM: destroy object operation is allowed on an object. This guideline makes OCM:destroy object operation recursive on containers. Any object that has bit 2 of the @dlna.dlnaManaged attribute true, will have the @restricted property set to "0".

7.4.1.7.14.4

[GUIDELINE] If a UPnP AV MediaServer supports the OCM: destroy object Operation and a control point invokes CDS:DestroyObject on a container with bit 2 of the @dlna.dlnaManaged attribute true and where any of that container's descendant objects have bit 2 of the @dlna.dlnaManaged attribute false then the following shall occur:

a) The UPnP AV Media Server shall destroy all descendent objects that have bit 2 of the @dlna.dlnaManaged attribute true, except those preserved by clause 3.

b) The UPnP AV Media Server shall not destroy an object that has bit 2 of the @dlna.dlnaManaged attribute false.

c) The UPnP AV Media Server shall not destroy any ancestor containers of an object from clause 2 even if those ancestor containers have bit 2 of the @dlna.dlnaManaged attribute true

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46] [49]	ZV8WT

NOTE: If an object exists within the subtree to be deleted that has bit 2 of the @dlna.dlnaManaged attribute false, it cannot be deleted. In order to preserve the containers between the root of the subtree and that object, all of the containers leading from the root to that object will be retained. Those containers leading from the root of the subtree to that object are the ancestor containers of that object.

7.4.1.7.14.5

[GUIDELINE] If a server encounters objects within the subtree that cannot be deleted because they have bit 2 of the @dlna.dlnaManaged attribute false, the server shall return a success response to a call of the CDS:DestroyObject that was initiated by the OCM: destroy object operation and SHALL behave according to 7.4.1.7.14.4

[Attributes]

M	R	DMS	M-DMS	n/a	[46] [49]	5BVYZ	

NOTE: The return of success is required by UPnP AV, the control point will need to perform a CDS:Browse() operation to determine if the entire subtree was deleted.

7.4.1.7.14.6

[Guideline] A control point shall not invoke a OCM: destroy object on an object if a bit 2 of the @dlna.dlnaManaged attribute is false for that object.

[Attributes]

M	R	+UP+ +UPSYNC+	M-DMU	n/a	[46] [49]	AMURA	

7.4.1.7.15 MM/CM: Use of Valid Values

[Guideline] A UPnP AV MediaServer control point shall always specify values for XML attributes and elements in a manner that conforms with the XML attribute's or element's schema.

[Attributes]

M	C	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	C8FTM	

NOTE: This is a general rule that applies whenever a control point creates or changes a metadata property.

7.4.1.7.16 M/CM: General Use of 7xx Error Codes

[Guideline] If a UPnP AV MediaServer responds to a CDS:CreateObject or CDS:UpdateObject request with a UPnP AV error code in the 700 - 799 range, then the UPnP AV MediaServer may use a localized, human-readable error message in the errorDescription tag of the SOAP response.

[Attributes]

O	A	DMS	M-DMS	n/a	[46] [49]	HMRW8	

NOTE: The ContentDirectory service specification does not provide adequate granularity for many error scenarios. This guideline allows DMS or M-DMS vendors to provide an error message that can be used by a user for error recovery and/or troubleshooting.

7.4.1.7.17 MM/CM: General Use of Error Code 720

[Guideline] Unless a different error code is mandatory, if a UPnP AV MediaServer receives a CDS action request for a particular upload AnyContainer or any OCM operation that is not supported or invalid, then the UPnP AV MediaServer may return a UPnP error code of 720 (Cannot process the request). The errorDescription tag in the SOAP response may contain a localized, human-readable error message.

[Attributes]

O	A	DMS	M-DMS	n/a	[46] [49]	8FTMQ	

NOTE: The ContentDirectory service specification does not provide adequate granularity for many error scenarios. As a workaround, DLNA allows implementations to return the error code 720 with a detailed error message. This error message can be used by a consumer for error recovery and/or troubleshooting.

7.4.1.7.18 MM/CM: Invalid 4th field Parameters

[Guideline] A UPnP AV MediaServer that receives a CDS:CreateObject that conflicts with the guideline in 7.4.1.3.12.10 may return a UPnP AV error code 712 (Bad Metadata) or it may return a success response and change the conflicting flag to an appropriate value.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[46] [49]	YSHAB

NOTE: In content upload usages, the DMS (not the Upload Controller) determines the transport layer options.

7.4.1.7.19 MM/CM: General Rule for Creating <res> Elements: Content Transfer Process

7.4.1.7.19.1

[GUIDELINE] If a UPnP AV MediaServer control point invokes CDS:CreateObject and specifies the creation of a <res> element that omits a URI value, then the control point shall specify a <res> element that conforms to the following rules.

- The first field of the res@protocolInfo value shall be "*".
- The second field of the res@protocolInfo value shall be "*".
- The third field of the res@protocolInfo value shall be a valid DLNA mime-type that correlates with the DLNA.ORG_PN value in the fourth field.
- The fourth field of the res@protocolInfo value shall have the DLNA.ORG_PN parameter and value. The value shall identify the DLNA media format profile of the content binary that will be used in the content transfer process. (It is permissible to include the ci-param.)

The fourth field of the res@protocolInfo value shall omit all other 4th field parameters defined by the DLNA guidelines, excluding DLNA.ORG_PN and DLNA.ORG_CI.

The <res> element shall omit the res@importUri attribute.

[ATTRIBUTES]

M	C	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	FB477

NOTE: A control point that specifies a <res> element without a URI value indicates that it wants to upload content to the MediaServer. A control point can deviate from the restrictions described in this guideline, but the result might be an error. For example, if the control point does not specify a DLNA.ORG_PN parameter, then the MediaServer can return an error.

7.4.1.7.19.2

[GUIDELINE] If a UPnP AV MediaServer creates a <res> element as a result of a CDS:CreateObject and the <res> element specified in the CDS request omits a URI value, then the created <res> element shall comply with the following rules.

- The <res> element shall omit a URI value.
- The <res> element shall have a res@importUri attribute and value. The res@importUri value shall indicate a URI that supports a content-transfer process. The length of the URI shall be less than or equal to 1024 bytes.

(It is permissible for the MediaServer to ignore or preserve a DLNA.ORG_CI parameter in the 4th field of the res@protocolInfo if the request specifies it.)

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	B4773

NOTE: If a MediaServer creates a <res> element intended to receive uploaded content, then it needs to provide a res@importUri attribute and value. This URI tells an Upload Controller which URI to perform an HTTP POST operation.

Note that the value of the <res> element needs to be empty until the content is actually available for serving as described in 7.4.1.7.27.4.

7.4.1.7.19.3

[GUIDELINE] If a UPnP AV MediaServer control point invokes CDS:CreateObject and specifies the creation of a <res> element without a URI value and the content is profiled as MPEG_PS_NTSC or MPEG_PS_PAL and it has discontinuous SCR and/or PTS, then the MediaServer control point shall specify a res@dlna:ifoFileURI attribute with an empty value.

The prefix for res@dlna:ifoFileURI shall be "dlna:" and the namespace shall be "urn:schemas-dlna-org:metadata-1-0/".

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	4773Q	

NOTE: Upload Controllers or M-DMUs might need to upload an IFO file. Providing an empty res@dlna:ifoFileURI signals the MediaServer to provide a URI value for it. In many cases, the MediaServer will include the res@dlna:importIfoFileURI attribute in the CDS:CreateObject response to indicate that the MediaServer will receive the IFO file. Upload Controllers and M-DMUs then perform a content transfer process to this URI in the same manner as a res@importUri.

7.4.1.7.19.4

[GUIDELINE] If a UPnP AV MediaServer creates a <res> element in response to a CDS:CreateObject request that has a res@dlna:ifoFileURI attribute with an empty value, then the UPnP AV MediaServer shall do one of the following to make the response.

- Preserve the res@dlna:ifoFileURI attribute with an empty value and add the res@dlna:importIfoFileURI attribute with a URI value that supports the content transfer process for the IFO file associated with the content binary. (Note that if a MediaServer implements this behavior, the MediaServer is permitted to generate a new content binary without discontinuities and generate a new IFO file for the new content binary. Similarly, a MediaServer is permitted to expose the uploaded content binary without modifications and generate a new IFO file.)
- Omit the res@dlna:ifoFileURI attribute and add the res@dlna:importIfoFileURI attribute with a URI value that supports the content transfer process for the IFO file associated with the content binary to indicate that the MediaServer will automatically generate an equivalent content binary without discontinuities after receiving both the IFO file and the content binary.
- Preserve the res@dlna:ifoFileURI attribute with an empty value and omit the res@dlna:importIfoFileURI attribute to indicate that the MediaServer will automatically create an IFO file (and expose it through the res@dlna:ifoFileURI) after a successful content transfer process on the res@importUri.
- Omit the res@dlna:ifoFileURI attribute and omit the res@dlna:importIfoFileURI attribute to indicate that the MediaServer will automatically generate an equivalent content binary without discontinuities after a successful content transfer process on the res@importUri.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	HABYL	

NOTE: Note that if a UPnP AV MediaServer will expose an IFO file, the res@dlna:ifoFileURI attribute remains an empty value until the MediaServer is ready to serve the IFO file, such as described in 7.4.1.7.27.7. If a CDS object has a <res> URI value for discontinuous SCR and/or PTS MPEG2, then URI values for <res> and res@dlna:ifoFileURI will be provided after the content transfer process. Ideally, both URIs are provided within 30 seconds of completing the content transfer process of the content binary, but some implementations can take longer (e.g. MediaServer performs validation or post-processing on the uploaded content binary.)

7.4.1.7.19.5

[GUIDELINE] If a UPnP AV MediaServer provides res@dlna:importIfoFileURI and a control point has intention to transmit a MPEG_PS_NTSC/PAL content and an associated IFO file to the MediaServer, then the control point shall completely transmit the IFO file before sending the MPEG_PS_NTSC/PAL content.

[ATTRIBUTES]

M	L	+UP+ +UPSYNC+	M-DMU	n/a	[46] [49]	FTMQ2	

NOTE: In the typical case, MediaServer uses the IFO file before receiving the content data to check its data size and boundary information in it.

7.4.1.7.19.6

[GUIDELINE] If a UPnP AV MediaServer control point invokes CDS:CreateObject and specifies the creation of a <res> element without a URI value, then the MediaServer control point should specify res@size attribute with a value equal to the byte length or equal to an estimated byte length of the content that will be sent during the content transfer process.

[ATTRIBUTES]

S	A	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	MRW8N	

NOTE: This allows a DMS or M-DMS to know how large the uploaded content will be and gives it the opportunity to reserve local storage space for the content.

Please note that the size of the content binary that is sent during the content transfer process can be different than the res@size value. For example, an Upload Controller can upload a content binary that is actually a conversion from another media format. In such a scenario, it can be very difficult to anticipate the exact size, so the Upload Controller specifies a res@size value that is a bit larger than the estimated size. During the content transfer process, the DMS will be able to determine the correct size of the content.

7.4.1.7.19.7

[GUIDELINE] If a UPnP AV MediaServer successfully completes a content transfer processs for the <res> element (and if res@size is present), then the MediaServer shall correct the res@size value to match the length of the content binary, if the indicated value does not match the byte length of the uploaded content binary.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	6TY85	

NOTE: Once content has been uploaded and the DMS or M-DMS can serve the content to other endpoints, the res@size value needs to be accurate. Estimated values will cause problems for Rendering Endpoints that rely on that metadata.

7.4.1.7.19.8

[GUIDELINE] If a UPnP AV Media Server which supports the upload AnyContainer operation or OCM: upload content operation for a content item profiled as MPEG_PS_NTSC/PAL receives a CDS:CreateObject request, it shall be able to accept the request regardless of whether it contains the res@dlna:ifoFileURI attribute.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	BN6RG	

NOTE: The UPnP AV Media Server can fail the CDS:CreateObject for normal exceptions such as unavailable resources

7.4.1.7.19.9

[GUIDELINE] If a UPnP AV MediaServer creates a <res> element in response to a CDS:CreateObject request that omits the res@dlna:ifoFileURI attribute, then the UPnP AV MediaServer shall omit the res@dlna:importIfoFileURI attribute to indicate that the MediaServer is not expecting a content transfer process for an IFO file.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	XNX4I	

NOTE: An IFO file can contain metadata not related to SCR and/or PTS discontinuities. A UPnP AV MediaServer can create an IFO file and expose it as described in subclause 7.4.1.4.8. The MediaServer's response can also omit the res@dlna:ifoFileURI attribute.

7.4.1.7.20 MM/CM: General Rule for Creating <res> Elements: Resume Content Transfer Process

7.4.1.7.20.1

[GUIDELINE] A UPnP AV MediaServer control point may support the resume content transfer operation. MediaServer control points may also make multiple resume content transfer operations.

The resume content transfer operation is a retry attempt for a failed content transfer process such that the content transfer starts from the point of transfer failure from a previous attempt.

[ATTRIBUTES]

O	A	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	Y4H43	

NOTE: Upload Controllers, Upload Synchronization Controllers, and M-DMUs are not required to support the ability to retry or resume a failed content transfer.

The DLNA guidelines provide interoperability rules for resume content transfer. The DLNA guidelines do not provide any normative mechanism for a retry attempt such that the content transfer process retries from the beginning of the content binary.

7.4.1.7.20.2

[GUIDELINE] If resume content transfer is supported, then retry IFO attempt shall also be supported.

For MediaServers, this means that returning a CDS:CreateObject response that has res@dlna:resumeUpload="1" shall mean that resume content transfer and retry IFO attempt are supported.

For MediaServer control point, this means that issuing a CDS:CreateObject request with res@dlna:resumeUpload="1" shall mean the MediaServer control point is capable of using resume content transfer and retry IFO attempt if an error occurs during the transmission of the content binary or IFO file.

A retry IFO attempt is defined as a retry attempt for a failed content transfer process attempt for an IFO file, such that the transfer begins from byte index 0 of the IFO file and the correspond HTTP POST request omits the CONTENT-Range HTTP header.

[ATTRIBUTES]

M	A	DMS +UP+ +UPSYNC+	M-DMS M-DMU	MIU	[46] [49]	6HEFW	

7.4.1.7.20.3

[GUIDELINE] If a UPnP AV MediaServer control point invokes CDS:CreateObject and wants to create a <res> element that supports the resume content transfer operation, then it shall specify a res@dlna:resumeUpload attribute with "1" value in a call to CDS:CreateObject.

The prefix for res@dlna:resumeUpload shall be "dlna:" and the namespace shall be "urn:schemas-dlna-org:metadata-1-0/".

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	4H438	

NOTE: Upload and Upload Synchronization Controllers create <res> elements with a res@dlna:resumeUpload="1" to determine if the MediaServer supports the resume content transfer operation. These Controllers will not specify res@dlna:resumeUpload="1" unless they intend to recover from a failed content transfer process by using the CONTENT-Range HTTP header in an HTTP POST request (i.e. using resume content transfer). The only other normative recovery mechanism for failed content uploads is to restart from a new upload AnyContainer or OCM: upload content operation.

7.4.1.7.20.4

[GUIDELINE] If a UPnP AV MediaServer creates a new <res> element as a result of a CDS:CreateObject request that specified the res@dlna:resumeUpload attribute with a value of "1" and if the MediaServer supports the resume content transfer operation for the created item, then the MediaServer shall preserve the value "1" for the res@dlna:resumeUpload attribute.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	HEFWQ	

7.4.1.7.20.5

[GUIDELINE] If the MediaServer creates the <res> element, it will confirm the support of the resume content transfer operation by preserving the res@dlna:resumeUpload="1" attribute and value.

A MediaServer respond res@dlna:resumeUpload="1" should keep 30 min the object when upload has failed.

If a UPnP AV MediaServer receives a CDS:CreateObject that specifies res@dlna:resumeUpload="0" or omits res@dlna:resumeUpload, then the MediaServer's CDS:CreateObject response shall specify res@dlna:resumeUpload="0" or omit res@dlna:resumeUpload.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	EFWQZ	

NOTE: MediaServers will never enable resume content transfer unless the Control Point requests that the feature apply to the upload operation because a control point can intentionally choose to rely on the required Auto-Destroy behavior when resume content transfer is not supported.

See 7.4.1.7.28.2 (MM/CM: Auto-Destroy Behavior for a Failed or Partial Content Transfer Process) for more information on MediaServer behavior when resume content transfer is not supported.

7.4.1.7.20.6

[GUIDELINE] If a UPnP AV MediaServer creates a new <res> element as a result of a CDS:CreateObject request that specified the res@dlna:resumeUpload attribute with a value of "1" and if the MediaServer cannot support the resume content transfer operation for the created item, then the MediaServer shall do one of the following.

- set the res@dlna:resumeUpload attribute to "0" in the created <res> element
- omit the res@dlna:resumeUpload attribute in the created <res> element

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	H4382	

NOTE: DMS implementations that do not support the resume content transfer operation will specify a value of "0" or omit the res@dlna:resumeUpload attribute to inform Upload and Upload Synchronization Controllers that the operation is not supported.

7.4.1.7.20.7

[Guideline] If a UPnP AV MediaServer control point attempts to recover from a failed content transfer process and both of the following conditions are true, then the Upload Controller or Upload Synchronization Controller shall use the resume content transfer operation as the initial recovery attempt.

- MediaServer control point specified res@dlna:resumeUpload="1" in a CDS request
- MediaServer preserved res@dlna:resumeUpload="1" in the CDS response that created the <res> element

[Attributes]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	OS33F	

NOTE: An M-DMU or Upload or Upload Synchronization Controller that specifies res@dlna:resumeUpload="1" will use an HTTP POST request with the CONTENT-Range HTTP header whenever possible. As a fallback, a Control Point is permitted to restart the upload process by restarting with an OCM: upload content operation. Some example scenarios using the fallback process are listed here.

- The CDS object or <res> element no longer exists on the DMS. (e.g. the suspend period is too long).
- The resume content transfer attempt has failed for some reason. In this case, a Control Point can use an OCM: destroy object operation before restarting with an OCM: upload content operation.

7.4.1.7.20.8

[Guideline] If a UPnP AV MediaServer control point attempts to recover from a failed content transfer process and at least one of the conditions are true, then the Control Point may attempt the recovery by starting over with an upload AnyContainer or OCM: upload content operation and follow up with a new content transfer process.

- MediaServer control point did not specify res@dlna:resumeUpload="1" in the CDS request that created the <res> element
- MediaServer control point received a CDS response with res@dlna:resumeUpload=0 or response omits res@dlna:resumeUpload attribute from the created <res> element

[Attributes]

O	A	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	ROS33	

NOTE: Usually the res@dlna:uploadedSize is omitted but it is not mandatory because it can simplify the MediaServer implementation. There is no normative use for this property in such scenarios.

7.4.1.7.21 MM res@dlna:uploadedSize

7.4.1.7.21.1

[Guideline] If a UPnP AV MediaServer supports the resume content transfer operation for a <res> element, then the <res> element shall have a res@dlna:uploadedSize attribute which has the same syntax and type as the res@size attribute.

The value of res@dlna:uploadedSize is the byte length of the content binary (not including the size of an uploaded IFO file) that was received during a content transfer process that failed.

The prefix for res@dlna:uploadedSize shall be "dlna" and the namespace shall be "urn:schemas-dlna-org:metadata-1-0/".

[Attributes]

M	A	DMS	M-DMS	n/a	[46] [49]	X4IVO	

NOTE: An M-DMU or Upload Controller can perform a resume content transfer by using an HTTP POST request with the CONTENT-Range header.

The res@dlna:uploadedSize does not represent the number of bytes that have been uploaded for an IFO file.

7.4.1.7.21.2

[GUIDELINE] If a UPnP AV MediaServer does not support the resume content transfer operation for a <res> element, then it may omit the res@dlna:uploadedSize attribute.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46] [49]	NX4IV	

NOTE: Usually the res@dlna:uploadedSize is omitted but it is not mandatory because it can simplify the MediaServer implementation. There is no normative use for this property in such scenarios.

7.4.1.7.22 MM/CM: General Rules for <res> Elements

7.4.1.7.22.1

[GUIDELINE] If a UPnP AV MediaServer creates a CDS object with a upnp:class value that is not equal to object.item.epgItem, object.item.video.videoBroadcast, object.item.audio.audioBroadcast, or any of their derived classes, then it shall have at least oneof the following URI values for each <res> element.

- URI value for the <res> element
- URI value for the res@importUri attribute

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46] [49]	6RGWV	C

NOTE: If both URI values are present, then the DMS indicates that uploading endpoints can overwrite the binary for a specific <res> element.

This guideline does not apply to CDS objects that represent EPG items or channel items, as these items can be available for streaming only at designated times, or might never be available for streaming at all.

7.4.1.7.22.2

[GUIDELINE] If a UPnP AV MediaServer creates a <res> element in response to a CDS:CreateObject, then the created <res> element may change the set of metadata attributes that were specified in the request.

[ATTRIBUTES]

O	L	DMS	M-DMS	n/a	[46] [49]	N6RGW	

NOTE: This guideline allows a control point to create <res> elements without knowledge of attributes supported by the MediaServer.

For example, a control point can specify a new <res> element with the res@size attribute but the created <res> element can omit res@size because the attribute is not supported.

Another example is when the DMS creates a <res> element that omits the res@dlna:ifoFileURI attribute when the request originally specified one. In this situation, the DMS created the <res> element because the DMS will create and serve an IFO file after it receives the movie file or the DMS will modify the movie file so that it does not have any discontinuities.

7.4.1.7.22.3

[GUIDELINE] If a <res> element of a CDS object with a upnp:class value that is equal to object.item.epgItem, object.item.video.videoBroadcast, object.item.audio.audioBroadcast, or any of their derived classes is not available for streaming, then a UPnP AV MediaServer shall omit the URI value from the <res> element.element.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	8YMDF	N

NOTE: In certain systems, a piece of content is available only at designated times. The URI pointing to the content will also only be valid at the designated times. A DMS will not expose the URI when the content is not available. Conversely, a DMS will expose the URI when the content becomes available.

7.4.1.7.22.4

[GUIDELINE] If a <res> element of a CDS object with a upnp:class value that is equal to object.item.epgItem, object.item.video.videoBroadcast, object.item.audio.audioBroadcast, or any of their derived classes is available for streaming, then a UPnP AV MediaServer shall include the URI value in the <res> element.element.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	VIUA9	N

NOTE:In certain systems, a piece of content is available only at designated times. The URI pointing to the content will also only be valid at the designated times. A DMS will not expose the URI when the content is not available. Conversely, a DMS will expose the URI when the content becomes available.

7.4.1.7.23 MM/CM: General Rules for CDS:CreateObject Request Syntax

7.4.1.7.23.1

[GUIDELINE] If a UPnP AV MediaServer control point invokes CDS:CreateObject, then it shall apply the mandatory metadata encoding rules for the following guidelines in the *Elements* input argument.

- 7.4.1.3.1 MM UPnP AV Control Point Tolerance of Unknown Property
- 7.4.1.3.2 MM DIDL-Lite Restrictions
- 7.4.1.3.3 MM DIDL-Lite Max Metadata Length
- 7.4.1.3.5 MM DIDL-Lite Boolean Values
- 7.4.1.3.6 MM upnp:class Values
- 7.4.1.3.7 MM DIDL-Lite dc:date Format
- 7.4.1.3.9 MM DIDL-Lite Desc Element Use
- 7.4.1.3.10 MM URI Rules
- 7.4.1.3.11 MM DIDL-Lite Recommended Metadata Properties
- 7.4.1.3.12 MM protocolInfo Context
- 7.4.1.3.15 MM DIDL-Lite protocolInfo values
- 7.4.1.3.16 MM protocolInfo values: 4th Field
- 7.4.1.3.17 MM pn-param (DLNA.ORG_PN Parameter)
- 7.4.1.3.22 MM ci-param (Conversion Indicator Flag)

[ATTRIBUTES]

M	L	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	Y855L	

NOTE: Control points are required to honor basic metadata restrictions that govern CDS metadata.

7.4.1.7.23.2

[GUIDELINE] If a UPnP AV MediaServer control point invokes CDS:CreateObject, then it shall apply the rules specified in 7.4.1.3.4.2 (in MM DIDL-Lite Non-empty Metadata Values) to all metadata, except the <res> element and @id attribute.

[ATTRIBUTES]

M	L	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	TY855

NOTE: The <res> element in a CDS:CreateObject is exempt from the non-empty value conventions because Upload Controllers do not specify the <res> URI value for some tasks, such as the upload AnyContainer or OCM: upload content operations.

7.4.1.7.23.3

[GUIDELINE] If a UPnP AV MediaServer control point invokes CDS:CreateObject, then it shall adhere to the following rules when specifying the *Elements* input argument.

- The XML shall exclude the <?xml> declarator and use UTF-8 encoding.
- The <DIDL-Lite> element shall be the top-most element.
- The <DIDL-Lite> element shall have a single <item> or a single <container> child element.
- At minimum, the <item> or <container> element shall specify the @id, @parentID, @restricted, dc:title, and upnp:class metadata properties.
- The @id value shall be an empty string ("").
- The @parentID value shall match the *ContainerID* input argument.
- The @restricted value shall be "0".
- The dc:title value shall comply with all DLNA-specified metadata restrictions.

Example request:

- CDS:CreateObject("10","<DIDL-Lite xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/" xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-Lite"> <item id="" restricted="0" parentID="10"> <dc:title>A picture in the park</dc:title> <upnp:class>object.item.imageItem</upnp:class> <res protocolInfo="*:*:image/jpeg:DLNA.ORG_PN=JPG_LRG"></res> </item> </DIDL-Lite>")

[ATTRIBUTES]

M	L	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	QBLX4

NOTE: This guideline specifies the baseline requirements for the metadata specified in a CDS:CreateObject request.

Control Points need to be aware that the created object can have metadata that is slightly different from what is sent. For example,

The @id value of the new object will be set.

The @parentID of the new object will be set (in the case where DLNA.ORG_AnyContainer is used).

Informative metadata, such as dc:title and dc:creator, can be truncated.

The @restricted value can be changed.

The returned object can include @dlna:dlnaManaged attribute.

upnp:class or upnp:createClass values can be changed to a derived or a base class value for audio, image, audio/video, and container objects.

7.4.1.7.23.4

[GUIDELINE] If a UPnP AV MediaServer control point invokes CDS:CreateObject, then it should provide all metadata defined in 7.4.1.3.11 for the given upnp:class.

[ATTRIBUTES]

S	C	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	SQBLX	

NOTE: Control points are encouraged to provide recommended metadata because it is useful for a user. If the MediaServer does not support the specified metadata, the CDS:CreateObject response will omit the unsupported metadata, as described in guideline.

7.4.1.7.23.5

[GUIDELINE] If a UPnP AV MediaServer control point invokes CDS:CreateObject, then it may specify additional metadata properties in a call to CDS:CreateObject, than those specified in 7.4.1.7.23.3.

[ATTRIBUTES]

O	R	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	ERZ5I	

NOTE: Control points are allowed to specify more than just the minimal metadata.

Note that control points are expected to handle a possible outcome described in 7.4.1.7.24.3.

7.4.1.7.23.6

[GUIDELINE] If a UPnP AV MediaServer control point invokes CDS:CreateObject with optional metadata, then the control point shall ensure that all necessary namespaces are properly declared.

[ATTRIBUTES]

M	R	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	SERZ5	

7.4.1.7.23.7

[GUIDELINE] A UPnP AV MediaServer that allows the creation of <dc:date> elements shall allow the control point to specify any valid form of <dc:date>, as defined by 7.4.1.3.7.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	W8N9T	

NOTE: This guideline allows the control point to specify any valid form for <dc:date>. Control points are not to assume that the format they specified in the request will be the final form of the <dc:date>.

7.4.1.7.23.8

[GUIDELINE] A UPnP AV MediaServer that allows the creation of <dc:date> elements may change the form of the <dc:date> value to match any valid form, as defined by 7.4.1.3.7.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[46] [49]	RW8N9	

NOTE: This guideline allows MediaServers the flexibility of using a consistent <dc:date> form for all of their CDS objects.

7.4.1.7.23.9

[GUIDELINE] If a UPnP AV MediaServer control point invokes CDS:CreateObject and the created object needs to support one or more OCM operations, then the MediaServer control point should specify the appropriate value for the @dlna:dlnaManaged attribute in the request.

[ATTRIBUTES]

S	A	+UP+ +UPSYNC+	M-DMU	MIU,	[46] [49]	MQ22G	

NOTE: For example, if a Control Point requires a created container to support the OCM: upload content operation, then the control point needs to specify a CDS container that supports the OCM: upload content operation in the CDS:CreateObject request by specifying the @dlna:dlnaManaged attribute with the appropriate bits set.

7.4.1.7.24 MM/CM: General Rules for CDS:CreateObject Response Syntax

7.4.1.7.24.1

[GUIDELINE] If a UPnP AV MediaServer that implements CDS:CreateObject, then it shall be able to support a request that specifies the properties specified in 7.4.1.7.23.3.

Example response:

- CDS:CreateObject("12","<DIDL-Lite xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/" xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/"> <item id="12" parentID="10" restricted="0"> <dc:title>A picture in the park</dc:title> <res importUri="http:/192.168.1.1/item?id=12" protocolInfo="*:*:image/jpeg:DLNA.ORG_PN=JPEG_LRG""/> <upnp:class>object.item.imageItem</upnp:class> <upnp:album>Photo Folder1</upnp:album> </item> </DIDL-Lite>")

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	BYLU6	

NOTE: Note that a UPnP AV MediaServer is not required to support the creation of container objects. In such cases, the proper behavior is to return UPnP error code 712, as described in guideline 7.4.1.7.23.2.

7.4.1.7.24.2

[GUIDELINE] If a UPnP AV MediaServer implements CDS:CreateObject, then it should accept the creation of recommended properties corresponding to the specified upnp:class, which are defined in the guideline 7.4.1.3.11.

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[46] [49]	TMQ22	

NOTE: Although not required, metadata such as dc:creator and upnp:album can be useful to a user.

7.4.1.7.24.3

[GUIDELINE] If a UPnP AV MediaServer creates a new CDS object in response to a CDS:CreateObject request (that conforms to DLNA guidelines) and the request specifies metadata properties that are unsupported by the MediaServer, then the MediaServer shall return a success response with the Result output that includes the metadata supported by the MediaServer.

Furthermore, if the MediaServer is able to return a success for a CDS:CreateObject request that specifies only the baseline metadata properties, then it shall also return a success for a CDS:CreateObject request that specifies the same baseline metadata properties and values with additional optional metadata properties.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46] [49]	ABYLU	

NOTE: The general approach for CDS:CreateObject is that a DMS needs to tolerate the presence of optional metadata properties (XML elements and attributes). The tolerance requirement also extends to 4th field parameters of protocolInfo, as required by 7.4.1.3.26.3 (MM other-param (Vendor-defined 4th field Parameters)).

The general exception to this approach is that a DMS is not required to tolerate <upnp:class> or <upnp:createClass> values that are unacceptable to a DMS. The primary reason for this exception is that derived classes can often imply a semantic difference. For example object.item.audioItem.audioBroadcast is not equivalent to an object.item.audioItem because the former implies that the stream is live. It is permissible for a DMS to tolerate a diversity of <upnp:createClass> and <upnp:class> values by automatically changing the specified values, but such behavior is not required.

A UPnP AV MediaServer that returns unsupported metadata in the Result but responds to CDS:Search and CDS:Browse without that metadata is behaving inconsistently. Returning with an error makes it difficult for control points to provide an information-rich experience. For this reason, DLNA requires that the CDS:CreateObject response includes only the metadata supported by the MediaServer. In this case, metadata supported by the MediaServer has the following characteristics.

- needs to minimally include the MediaServer's supported metadata properties (XML elements and attributes) that was specified in the request,
- and can optionally include additional metadata properties (XML elements and attributes) that the MediaServer added as part of the success CDS:CreateObject response.

7.4.1.7.24.4

[GUIDELINE] A UPnP AV MediaServer that creates a new CDS object, in response to a CDS:CreateObject request, may add additional metadata properties (XML elements or attributes) in the CDS:CreateObject response.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[46] [49]	73QLR	

NOTE: The following are some examples of when a DMS might add additional metadata properties.

- The DMS adds the @dlna:dlnaManaged attribute to indicate the supported OCM operations for a created CDS object.
- The DMS adds one or more upnp:createClass elements (in addition to those specified in the CDS:CreateObject request) to indicate additional objects that can be created in a new CDS container.

7.4.1.7.24.5

[GUIDELINE] A UPnP AV MediaServer that receives a CDS:CreateObject request may change the value of the @restricted attribute.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46] [49]	3QLRK	

7.4.1.7.24.6

[GUIDELINE] If a UPnP AV MediaServer that indicates support for OCM operations (as defined in guideline 7.4.1.7.8.4) receives a CDS:CreateObject request that specifies the creation of a CDS object with support for one or more OCM operations that cannot be supported for the specified object, then the MediaServer shall return a failure with UPnP AV error code 712 (Bad Metadata).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	773QL	

7.4.1.7.24.7

[GUIDELINE] If a UPnP AV MediaServer that indicates support for OCM operations (as defined in guideline 7.4.1.7.8.4) receives a CDS:CreateObject request, it may change the value of the @dlna:dlnaManaged attribute as long as the new value indicates a superset of the requested OCM operation.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46] [49]	EY8ZB	

NOTE: DMS or MDMS devices are always allowed to increase the set of supported operations on a created object. Otherwise, the DMS is required to return an error when the requested OCM operations cannot be provided for the CDS object described in the CDS:CreateObject request.

7.4.1.7.24.8

[GUIDELINE] If a UPnP AV MediaServer that does not indicate support for any OCM operations (as defined in guideline 7.4.1.7.8.4) receives a CDS:CreateObject request that specifies the creation of a CDS object with support for one or more OCM operations, and if the server creates a CDS object as a result of the CDS:CreateObject request, it shall ignore the @dlna:dlnaManaged attribute in the request and omit the @dlna:dlnaManaged attribute from the created object.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	8Y9WS	

NOTE: DMS or M-DMS devices that do not support OCM operations might not understand @dlna:dlnaManaged attribute, and are required to drop the OCM flags from the created object regardless of the validity of the attribute in the request.

7.4.1.7.25 MM/CM: General Rules for CDS:CreateObject Errors

7.4.1.7.25.1

[GUIDELINE] If a UPnP AV MediaServer rejects a CDS:CreateObject request due to invalid metadata values in the request, then the MediaServer shall return UPnP AV error code 712 (Bad metadata).

Invalid values are values that are not valid for the data type (as defined by the appropriate schema) or values that are not supported by the MediaServer implementation.

Valid, non-empty values are values that are valid for the data type specified for the metadata property and also not composed entirely of white space characters.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46] [49]	QLRK8	

NOTE: A UPnP MediaServer might choose to accept a CDS:CreateObject request despite invalid metadata values. For example, a MediaServer that always replaces a certain metadata value with a preset value can simply ignore that metadata value contained in the request.

Note that a UPnP AV MediaServer that removes the entire metadata property from the Result output is informing the control point that the specified property is not supported, rather than informing the control point that the value is invalid. This is described in 7.4.1.7.24.3.

Although this guideline applies to the MediaServer, control points are expected to provide valid metadata values, as required by 7.4.1.7.15 and 7.4.1.7.24.

7.4.1.7.25.2

[GUIDELINE] If a UPnP AV MediaServer returns a CDS:CreateObject response with a UPnP error code, it shall not result in the creation of a new CDS object.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46] [49]	YLU6X	

7.4.1.7.25.3

[GUIDELINE] If the UPnP AV MediaServer does not support the CDS:CreateObject action because the specified upnp:class or res@protocolInfo is not acceptable to the MediaServer, it shall respond with an error code 712 (Bad Metadata).

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46] [49]	LU6X9	

7.4.1.7.25.4

[GUIDELINE] If a UPnP AV MediaServer receives a CDS:CreateObject request with multiple <res> elements and the UPnP AV MediaServer does not allow the creation of objects with multiple <res> elements, then it shall return a UPnP AV error code of 712 (Bad Metadata).

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	Q22GR	

NOTE: MediaServers are not required to support multiple <res> elements.

7.4.1.7.25.5

[GUIDELINE] If a UPnP AV MediaServer rejects a CDS:CreateObject request because the <res> element specifies a URI value or a res@importUri value, then it shall return a UPnP AV error code of 712 (Bad Metadata).

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	22GRN	

7.4.1.7.25.6

[GUIDELINE] If a UPnP AV MediaServer cannot accept a CDS:CreateObject request due to the processing capacity or current state of the device, then it shall respond with error code 720 (Cannot process the request).

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46] [49]	8N9TA	

7.4.1.7.25.7

[GUIDELINE] If a UPnP AV MediaServer cannot accept a CDS:CreateObject request due to the lack of storage capacity, then the MediaServer shall return a UPnP error response with error code of 720 (Cannot process the request).

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46] [49]	N9TA4	

NOTE: A MediaServer can return this error code as a result of a res@size value specified in a CDS:CreateObject request. A DMS can also return this value when the upload capacity for the DMS has been reached.

7.4.1.7.25.8

[GUIDELINE] If a UPnP AV MediaServer receives a CDS:CreateObject request with zero <res> elements and the UPnP AV MediaServer does not allow the creation of objects with zero <res> elements, then it shall return a UPnP AV error code of 712 (Bad Metadata).

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	3Y33M	

NOTE: This behavior is conditional mandatory because a UPnP AV MediaServer might be able to support the creation of a container object with zero <res> elements.

7.4.1.7.26 MM/CM: Content Transfer Process

7.4.1.7.26.1

[GUIDELINE] If a UPnP AV MediaServer control point is going to initiate a content transfer process for the first file (either the IFO file or the actual content binary) of a CDS object, then it shall do so within 30 seconds of the last event described below.

- The CDS:CreateObject response from a Upload AnyContainer or an OCM: upload content operation

If a UPnP AV MediaServer control point uploads an IFO file, then the IFO file shall be uploaded first and the transfer of the actual content binary shall be started within 30 seconds of completing the IFO file transfer.

[ATTRIBUTES]

M	L	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	RZ5I6	

NOTE: Within 30 seconds of creating the CDS object, the Control Point begins its first content transfer for the CDS object.

7.4.1.7.26.2

[GUIDELINE] A UPnP AV MediaServer control point shall utilize the URI provided in a res@importUri or a res@dlna:importIfoFileURI as the destination URI for a content transfer process.

UPnP AV MediaServer control points shall only upload content binaries to the res@importUri.

UPnP AV MediaServer control points shall only upload IFO files to the URI value of res@dlna:importIfoFileURI.

[ATTRIBUTES]

M	L	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	Z5I6Y	

NOTE: The CDS specification is ambiguous as to whether a <res> element's URI value can be used to overwrite content. These guidelines limit the control to using import URI values for both initial and overwriting content transfers. See 7.5.4.3.6.1 for HTTP client implications.

Note that IFO files are not considered content binaries, so uploading an IFO file to a res@importUri file is expressly prohibited.

For additional information on performing a content transfer process for either a content binary or an IFO file, see 7.5.4.3.6.1.

For additional information on creating a <res> element for uploading IFO files, see 7.4.1.7.19.3 (MM/CM: General Rule for Creating <res> Elements that Support a Content Transfer Process).

7.4.1.7.26.3

[GUIDELINE] If a UPnP AV MediaServer control point fails to completely upload an IFO file and if res@dlna:resumeUpload="1" and if the control point initiates a retry IFO attempt, then the control point shall initiate the retry IFO attempt within 30 minutes of the failure.

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	BLX49	

NOTE: Retry attempts to transfer the IFO file needs to start within 30 minutes of the time of failure because the MediaServer will destroy the CDS object after 35 minutes has elapsed since the failure.

7.4.1.7.26.4

[GUIDELINE] If a UPnP AV MediaServer control point fails to completely upload a content binary and if res@dlna:resumeUpload="1" and if the control point initiates a resume content transfer, then the control point shall initiate the resume content transfer within 30 minutes of the failure.

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	LX49K	

NOTE: Resume attempts to transfer the content binary file needs to start within 30 minutes of the time of failure because the MediaServer will destroy the CDS object after 35 minutes has elapsed since the failure.

7.4.1.7.26.5

[GUIDELINE] A UPnP AV MediaServer control point shall not attempt to use retry IFO transfer or resume content transfer unless res@dlna:resumeUpload="1".

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	855LC	

7.4.1.7.26.6

[GUIDELINE] A UPnP AV MediaServer shall facilitate a content transfer process by supporting an HTTP POST request on a URI for a res@importUri or a res@dlna:importIfoFileURI.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	55LC5	

NOTE: DLNA defines HTTP POST as the only mechanism for a content transfer process. DLNA might define additional mechanisms for a content transfer process in the future.

7.4.1.7.27 MM/CM: General Rules After a Successful Content Transfer Process

7.4.1.7.27.1

[GUIDELINE] If a UPnP AV MediaServer successfully completes a content transfer process, then the MediaServer may keep or remove the res@importUri attribute.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[46] [49]	RGWVX	

NOTE: A MediaServer that keeps the res@importUri attribute is informing the network that an Upload Controller can overwrite the content that was uploaded.

7.4.1.7.27.2

[GUIDELINE] If a UPnP AV MediaServer keeps the res@importUri attribute and creates a URI value for the <res> element (to signal that the Content Source will serve the URI), then the MediaServer may specify the same URI value for both the res@importUri and the URI value of a <res> element.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[46] [49]	4IVOE

NOTE: MediaServers are permitted to use the same URI for both the <res> URI value and the res@importUri value. In such cases, MediaServers need to ensure that the <res> value is not exposed until after the content has been received, as described in guideline 7.4.1.7.27.4.

7.4.1.7.27.3

[GUIDELINE] If a UPnP AV MediaServer keeps the res@dlna:ifoFileURI attribute and creates a res@dlna:importIfoFileURI value for the <res> element (to signal that the Content Source will serve the URI), then the MediaServer may specify the same URI value for both the res@dlna:importIfoFileURI and the res@dlna:ifoFileURI value of a <res> element.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[46] [49]	GWVX7

NOTE: MediaServers are permitted to use the same URI for both the <res> URI value and the res@importUri value. In such cases, MediaServers need to ensure that the <res> value is not exposed until after the content has been received, as described in guideline 7.4.1.7.27.4.

7.4.1.7.27.4

[GUIDELINE] If a UPnP AV MediaServer is going to serve uploaded content to other endpoints, then the DMS shall provide the URI value of the <res> element only after the DMS can serve the content.

Example of <res> before content is uploaded to the DMS:

- <res protocolInfo="http-get:*:image/jpeg:*:DLNA.ORG_PN=JPEG_LRG" importUri="http://192.168.1.10/upload?file=content-data-1234.jpg"/>

Example of <res> after content is uploaded to the DMS:

- <res protocolInfo="http-get:*:image/jpeg:*:DLNA.ORG_PN=JPEG_LRG"> http://192.168.1.10/content-data-1243.jpg</res>

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46] [49]	IVOE5

NOTE: A UPnP AV MediaServer that advertises a <res> URI value before the content can actually be served is making a false claim that the content is available.

Note that a DMS that serves partially uploaded content is responsible for conforming to the guidelines related to the media format profile.

7.4.1.7.27.5

[GUIDELINE] A UPnP AV MediaServer may automatically create additional <res> elements (with network-accessible URI values and protocolInfo values) after a successful content transfer process.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[46] [49]	S33FG	

NOTE: MediaServers might want to provide additional <res> elements to support content transformations or additional transport protocols.

7.4.1.7.27.6

[GUIDELINE] If a UPnP AV MediaServer is going to serve uploaded content that has an IFO file, then the MediaServer shall provide the URI value of the <res> element after the Content Source can provide the IFO file.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	33FGH	

NOTE: Advertising content with SCR/PTS discontinuities without providing an IFO file is a violation of guideline 7.4.1.4.8.2.

7.4.1.7.27.7

[GUIDELINE] If a UPnP AV MediaServer is going to serve uploaded content that has an IFO file, and successfully receives an IFO file in a content transfer process, then the MediaServer shall provide a res@dlna:ifoFileURI attribute with a URI value for the IFO file. Until the MediaServer has the IFO file, the res@dlna:ifoFileURI shall have an empty value.

Example timeline (with additional comments for other hypothetical situations after ***) Numbers in angle brackets, <n>, refer to steps in this timeline.

a) 1. Upload Controller or M-DMU invokes CDS:CreateObject with a <res> element for MPEG_PS_NTSC. The request includes res@dlna:ifoFileURI="" to indicate that it will also upload an IFO file. The request also includes res@dlna:resumeUpload="1" to indicate that the Upload Controller will use resume features if a failure occurs during a transfer. *** If the Upload Controller / M-DMU is not uploading an IFO file, it would have omitted res@dlna:ifoFileURI. If the Upload Controller / M-DMU was not going to use resume, it would have omitted res@dlna:resumeUpload.

b) 2. The DMS or M-DMS responds with success. The response includes a res@importUri and res@dlna:importIfoFileURI, with URI values that will receive the uploaded content. The response preserves the res@dlna:resumeUpload="1" to indicate resume is supported. *** If the DMS/M-DMS does not support resume, it would have omitted res@dlna:resumeUpload or set the value to "0". A DMS/M-DMS never returns res@dlna:resumeUpload="1" unless requested to do so.

c) 3. The Upload Controller begins uploading the IFO file.

d) 4. During the transfer from <3>, an error occurs and the transfer fails.

e) 5. Within 30 minutes of the error in <4>, the Upload Controller or M-DMU issues an HTTP POST to retry the transfer of the IFO file. The Upload Controller/M-DMU does not use CONTENT-Range when retrying an IFO transfer. *** If the Upload Controller/M-DMU does not do the retry, then the DMS/M-DMS would automatically destroy the new CDS object (from <1>). The DMS/M-DMS waits at least 35 minutes from the failure time in <4> before automatically destroying. In scenarios where resume is not enabled, the DMS/M-DMS would destroy the CDS object (from <1>) at any time after the error in <4> occurred.

f) 6. After completing the retry IFO transfer from <5>, the Upload Controller/M-DMU initiates the transfer of the content binary within 30 seconds. *** If the Upload Controller/M-DMU is not uploading an IFO, then it would have started the content transfer within 30 seconds of receiving the CDS:CreateObject response.

g) 7. During the transfer from <6>, an error occurs and the transfer fails.

h) 8. Within 30 minutes of the error in <7>, the Upload Controller or M-DMU issues an HTTP POST with the CONTENT-Range header to resume the transfer from the byte position where the failure occurred. *** If the Upload Controller/M-DMU does not do the resume, then the DMS/M-DMS would automatically destroy the new CDS object (from <1>). The DMS/M-DMS waits at least 35 minutes from the failure time in <7> before automatically destroying. In scenarios where resume is not enabled, the DMS/M-DMS would destroy the CDS object (from <1>) at any time after the error in <7> occurred.

i) 9. The content transfer from <6> succeeds and the DMS/M-DMS advertises URI values for the <res> element and the res@ifoFileURI.

j) 10. The DMS or M-DMS also automatically creates a new <res> element for its locally converted version of MPEG_PS_NTSC_XAC3

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	43829

NOTE: Just as a MediaServer uses the <res> URI value to indicate that the content is available for Content Receivers, the MediaServer also uses the res@dlna:ifoFileURI attribute to indicate that the IFO file is available for Content Receivers.

7.4.1.7.28 MM/CM: Auto-Destroy Behavior for a Failed or Partial Content Transfer Process

7.4.1.7.28.1

[GUIDELINE] If a UPnP AV MediaServer implements CDS:CreateObject for use with the listed operations, then it shall implement Auto-Destroy behavior.

Listed operations:

- upload AnyContainer
- OCM: upload content

The Auto-Destroy Behavior shall be defined as removing the created CDS object from the CDS hierarchy exposed by the MediaServer, such that the CDS object cannot be found when browsing or searching the MediaServer.

The time when the CDS object is actually removed from the CDS hierarchy is not entirely defined by the guidelines. However, other guidelines in 7.4.1.7.28 have restrictions when the CDS object is automatically removed.

All guidelines in 7.4.1.7.28 assume Ideal Network Conditions and interoperability is not guaranteed in scenarios involving user-initiated or out-of-scope system events.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	FWQZZ

NOTE: A DMS that supports the Upload System Usage needs to automatically destroy CDS objects when the files are not properly received by the MediaServer.

The guidelines guarantee interoperability only under Ideal Network Conditions. Scenarios that involve user-initiated events or out-of-scope system events do not guarantee interoperability. Examples include:

- Another control point invoking CDS:DestroyObject causes a failure during an upload AnyContainer operation.
- The MediaServer product canceling all active uploads because it is starting to record a video broadcast to local storage.

The actual time when the MediaServer destroys such CDS items is not defined by DLNA. Some examples include a 35-minute timer mechanism, a clean-up mechanism that executes when the device has been idle for an extended period of time, or a cleanup operation that only executes when certain CDS actions are called.

7.4.1.7.28.2

[GUIDELINE] If res@dlna:resumeUpload="0" or res@dlna:resumeUpload is omitted, a UPnP AV MediaServer shall perform Auto-Destroy behavior on CDS objects (created through upload AnyContainer or OCM: upload content) under these conditions:

- 35 seconds has elapsed since the CDS:CreateObject response was completely sent and the first content transfer process (for either the IFO file or the content binary) has not started
- 35 seconds has elapsed since the CDS object's IFO was completely received and the content transfer process for the content binary has not started
- The content transfer process of the content binary or the IFO file has failed.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	38293

NOTE: This guideline explains when a MediaServer needs to automatically destroy CDS objects that were created from upload AnyContainer or OCM: upload content.

The guideline explicitly covers the case where resume content transfer, as described in 7.4.1.7.20, is not supported.

7.4.1.7.28.3

[GUIDELINE] If res@dlna:resumeUpload="1", a UPnP AV MediaServer shall perform Auto-Destroy behavior on CDS objects (created through upload AnyContainer or OCM: upload content) under these conditions:

- 35 seconds has elapsed since the CDS:CreateObject response was completely sent and the first content transfer process (for either the IFO file or the content binary) has not started
- 35 seconds has elapsed since the CDS object's IFO was completely received and the content transfer process for the content binary has not started
- 35 minutes has elapsed since the failure of a content transfer process for the CDS object's IFO and retry IFO attempt has not started
- 35 minutes has elapsed since the failure of a content transfer process for the CDS object's content binary and resume content transfer process has not started

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	WQZZF

NOTE: This guideline explains when a MediaServer needs to automatically destroy CDS objects that were created from upload AnyContainer or OCM: upload content.

The guideline explicitly covers the case where resume content transfer, as described in 7.4.1.7.20, is supported.

7.4.1.7.28.4

[GUIDELINE] A UPnP AV MediaServer shall not automatically destroy a CDS object or any of its <res> element when the MediaServer is doing a content transfer process for the CDS object.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	QZZFP

This guideline requires MediaServer implementations not to destroy CDS objects or <res> elements involved in an Upload System Usage.

As implied from the comment for 7.4.1.7.28.1, a MediaServer might choose to destroy such an object as a result of a CDS:DestroyObject request that was invoked from another control point. This type of a scenario is an unexpected user-initiated and is outside the scope of the Ideal Network Conditions assumption.

7.4.1.7.28.5

[GUIDELINE] If a UPnP AV MediaServer partially completes a content transfer process for a <res> element and res@dlna:resumeUpload="1", then the MediaServer shall keep all res@importUri and res@dlna:importIfoFileURI that are associated with the CDS object valid for at least another 35 minutes.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	293RE

NOTE: This guideline clarifies that MediaServers needs to preserve the validity of URI values that are used for uploaded IFO files and content binaries.

After a MediaServer partially receives a content binary, the device needs to extend the validity of the URI values (that have not completed a content transfer process) for at least 35 minutes. This provides a reasonable amount of time for an Upload Controller to retry or set up the next content transfer process.

See 7.4.1.7.20.4 - 7.4.1.7.20.8 and 7.5.4.3.6.3 for more information about resuming a failed content transfer process.

7.4.1.7.28.6

[GUIDELINE] A UPnP AV MediaServer control point may use an OCM: destroy object operation to cancel the following operations: upload AnyContainer or OCM: upload content.

[ATTRIBUTES]

O	C	+UP+ +UPSYNC+	M-DMU	MIU	[46] [49]	ZZFPW

NOTE: A Control Point uses this operation to signal the DMS to undo its previous operations that resulted in the creation of a CDS object.

7.4.1.7.29 MM/CM: Content Validation and Advertisement

7.4.1.7.29.1

[GUIDELINE] If a UPnP AV MediaServer successfully completes a content transfer process for a CDS resource that was created with the DLNA.ORG_PN parameter in the 4th field, then the MediaServer may advertise that content using the supplied DLNA media format profile.

Advertise that content is meant specifically that the MediaServer provides a URI value for the <res> element. Before the MediaServer receives the content binary, the <res> element (without a URI value) will be advertised along with the res@importUri.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46] [49]	8293R

NOTE: DLNA UPnP AV MediaServers that support the Upload Device Option can assume that control points that claim content conforms to DLNA are actually going to upload DLNA-conformant content. However it is still good practice for a UPnP AV MediaServer to examine the uploaded content against the profile definition before advertising the content as being DLNA conformant.

7.4.1.7.29.2

[GUIDELINE] If a UPnP AV MediaServer receives a content binary for a CDS resource that does not have the DLNA.ORG_PN parameter and it intends to advertise the content, then the MediaServer shall validate the content before advertising it with the DLNA.ORG_PN parameter.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46] [49]	FGH9D	

NOTE: Although an algorithm for validation is not specified, a UPnP AV MediaServer that determines that uploaded content is conformant to a DLNA media format profile is permitted to advertise it as such. The guideline does not obligate a DMS to either support uploading or advertising of non-DLNA content.

7.4.1.7.29.3

[GUIDELINE] A UPnP AV MediaServer may change the value of a res@protocolInfo as part of the content validation process.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[46] [49]	3FGH9	

7.4.1.7.29.4

[GUIDELINE] If a UPnP AV MediaServer exposes uploaded content, then it should examine the uploaded content binaries to ensure that the metadata reported through the CDS is consistent with the content binary. In cases where a conflict is found, a UPnP AV MediaServer should appropriately correct the metadata reported through CDS.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46] [49]	OE52V	

NOTE: This is a good behavioral practice and helps to prevent interoperability problems.

7.4.1.7.30 MM/CM: General Rules for CDS:DestroyObject

[GUIDELINE] If a UPnP AV MediaServer implements CDS:DestroyObject, then a CDS:DestroyObject request on a CDS item, with the @dlna:dlnaManaged attribute, shall result in the complete removal of the CDS item.

If the CDS:DestroyObject is a success, then the CDS item is removed from the CDS hierarchy.

If the CDS:DestroyObject is a failure, then the CDS item remains unaltered.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46] [49]	VOE52	

NOTE: Partially destroying a CDS item (such as some metadata or <res> elements) is problematic because control points have no way to recover the partially lost information.

The related question of whether the actual content binaries are deleted from the local storage is vendor-defined.

Developers, note that this guideline applies only to CDS Content Items (object.item) and not CDS Containers (object.container).

7.4.2 Image Printing Media Management

7.4.2.1 General

This subclause of the DLNA Home Networked Device Interoperability Guidelines covers the guidelines for implementing image printing media management using the UPnP Printer architecture. Annex D also provides much useful information for implementers that want to implement the printing-related usages.

7.4.2.2 General Capability Requirements

7.4.2.2.1 MM UPnP Printer Compliance

[GUIDELINE] DLNA Device Classes shall fully support the mandatory portions of the UPnP Printer:1 specifications.

[ATTRIBUTES]

M	R	DMPr	n/a	n/a	[50] [88] [89] [90]	VX7W7	

NOTE: UPnP Printer:1 is the baseline architecture for discovering image printer devices.

7.4.2.3 Device Requirements

7.4.2.3.1 MM UPnP Printing Controller-1 Definition

7.4.2.3.1.1

[GUIDELINE] A DLNA Device Class may implement the Printing Controller-1 Device Capability.

[ATTRIBUTES]

O	A	DMS DMP DMR DMC DMPr	M-DMS M-DMP M-DMC M-DMD M-DMU	n/a	n/a	AX58Z	N

7.4.2.3.1.2

[GUIDELINE] The +PR1+ Device Capability shall implement a UPnP Printer control point that uses the PE:CreateURIJob action to supply print jobs that allows selected image content to be printed.

[ATTRIBUTES]

M	L	+PR1+	n/a	n/a	[50] [88] [89] [90]	WVX7W	

NOTE: This guideline indicates that the +PR1+ Device Capability will use a UPnP control point that controls a UPnP Printer to print images it has locally.

+PR1+ is recommended to delete the XHTML file after its use. Please refer to 7.4.2.3.5.

7.4.2.3.2 MM UPnP Printing Controller-2 Definition

7.4.2.3.2.1

[GUIDELINE] A DLNA Device Class may implement the Printing Controller-2 Device Capability.

[ATTRIBUTES]

O	A	DMS DMP DMR DMC DMPr	M-DMS M-DMP M-DMC M-DMD M-DMU	n/a	n/a	4YFJX	N

7.4.2.3.2.2

[GUIDELINE] The +PR2+ Device Capability shall implement a UPnP Printer control point that uses the PE:CreateURIJob action to supply print jobs that allows selected image content to be printed.

[ATTRIBUTES]

M	L	+PR2+	n/a	n/a	[50][50] [88] [89] [90]	LC5BH	

NOTE: This guideline indicates that the +PR2+ Device Capability will use a UPnP control point that controls a UPnP AV MediaServer to find image content and a UPnP Printer to print those images found on a UPnP AV MediaServer.

+PR2+ is recommended to delete the XHTML file after its use. Please refer to 7.4.2.3.5.

7.4.2.3.2.3

[GUIDELINE] The +PR2+ Device Capability shall implement a UPnP AV MediaServer control point for browsing a ContentDirectory service on a DMS or M-DMS to obtain image content to be printed.

[ATTRIBUTES]

M	R	+PR2+	n/a	n/a	[46] [49]	49K78	

7.4.2.3.3 MM DMPr UPnP Printer Device Definition

7.4.2.3.3.1

[GUIDELINE] A DMPr shall implement a UPnP Printer:1 device that shall haveone PrintBasic:1 service and one PrintEnhanced:1 service.

[ATTRIBUTES]

M	R	DMPr	n/a	n/a	[88] [89] [90]	5LC5B	C

NOTE: DMPr devices need to implement the minimum baseline services for a UPnP Printer. Although the UPnP definition for a Printer:1 requires PrintBasic:1 and PrintEnhanced:1, the DLNA guidelines only define interoperability for the PrintEnhanced:1 service. In most cases, the implementation for a PrintBasic:1 service is the same underlying implementation used for a PrintEnhanced:1 service since PrintEhhanced:1 is a superset of PrintBasic:1.

7.4.2.3.3.2

[GUIDELINE] A UPnP Printer shall identify in the Device Description Document the PrintBasic:1 service and the PrintEnhanced:1 service using serviceType and serviceId elements with the following values:

Table 29 — DMPr serviceType and serviceId Values

Service	Element	Value
PrintBasic service	serviceType	urn:schemas-upnp-org:service:PrintBasic:1
	serviceID	urn:upnp-org:serviceId:1
PrintEnhanced service	serviceType	urn:schemas-upnp-org:service:PrintEnhanced:1
	serviceID	urn:upnp-org:serviceId:3

[ATTRIBUTES]

M	C	DMPr	n/a	n/a	[88] [89] [90]	H6BYD	N

NOTE: These serviceType and serviceId values uniquely identify the type of service. The table above lists the element values in compliance with the related UPnP specifications.

7.4.2.3.4 MM DMPr PrintEnhanced Rules

[GUIDELINE] A DMPr shall fully support the mandatory actions and state variables for a UPnP PrintEnhanced:1 service.

[ATTRIBUTES]

M	R	DMPr	n/a	n/a	[90]	X49K7

NOTE: PrintEnhanced:1 adds several mandatory features over PrintBasic:1 that enables better photo printing capabilities such as the ability to print borderless photos.

7.4.2.3.5 MM XHTML-Print

7.4.2.3.5.1

[GUIDELINE] A UPnP Printer device shall specify the <dlna:X_DLNACAP> element (as a child of the <device> element that represents the UPnP Printer device) that includes the printProfiles-AllRangeTrue-AllRangeFalse token.

Table 30 — UPnP Printer dlna:X_DLNACAP Element

Capability ID	Description
printProfiles- AllRangeTrue - AllRangeFalse	ndicates the supported DLNA XHTML-Print document profiles for these scenarios : • the XHTML-Print document only references images that support the HTTP RANGE header • • the XHTML-Print document references one or more images that do not support the HTTP RANGE header

More formally, the syntax of the capability ID is defined below.

- printer-capability-id = "printProfiles" "-" all-range-true-profile "-" all-range-false-profile
- all-range-true-profile = <DLNA XHTML profile name>
- all-range-false-profile = <DLNA XHTML profile name>

The "printProfiles" portion of the capability ID is a literal, case-sensitive string value.

The all-range-true-profile is the top-most XHTML profile that the Printer device supports when all images referenced by the XHTML-Print document support the HTTP RANGE header. The all-range-true-profile value shall be XHTML_PT or a superset XHTML profile.

The all-range-false-profile is the top-most XHTML profile that the Printer device supports when one or more images referenced by the XHTML-Print document do not support the HTTP RANGE header. The all-range-false-profile value shall be equal to XHTML_Baseline or a superset XHTML profile.

Given the hierarchy of XHTML profiles defined in 10.2 in [56], the top-most XHTML profile is the profile that appears highest in the hierarchy, closest to the root XHTML profile of XHTML_ALL.

A UPnP printer device that supports a DLNA-defined XHTML profile shall adhere to these rules.

- The Printer device shall be able to properly print any XHTML-Print document that conforms to the indicated XHTML profile definition.
- The Printer device shall be able to properly print any XHTML-Print document that conforms to an XHTML profile that is a subset of the indicated XHTML profile.
- The Printer device shall be able to properly print any of the photo templates contained in [54] within the printers range of supported paper sizes and borderless capability.
- Properly print is defined as the ability to print the XHTML-Print document in a manner consistent with all mandatory portions of [52] and [53] including but not limited to the document's instructions for image placement, image rotation, and borders.
- The definition of properly print imposes these additional requirements on the content binary lengths (a.k.a. image file sizes).
  - If the XHTML profile is XHTML_Baseline and the referenced image does not support the HTTP RANGE header, then the Printer device shall be able to print the image, only when its content binary length is equal to or less than 4194304 bytes (4 MB).
  - Otherwise, the Printer device shall be able to print each image, so long as the image meets all restrictions imposed by an image profile that is defined for use by the XHTML profile definition. This restriction applies regardless of whether the images support the HTTP RANGE header or not.

The definition of support the HTTP RANGE header is defined by whether the op-param's b-val is "1", as obtained from:

- the image's 4th field value of res@protocolInfo value (acquired from CDS), and
- the contentFeatures.dlna.org value (acquired from an HTTP transport layer request that specified getcontentFeatures.dlna.org)

Example:

```
<dlna:X_DLNACAP>
printProfiles-XHTML_Complex-XHTML_Baseline
</dlna:X_DLNACAP>
```

[ATTRIBUTES]

M	A	DMPr	n/a	n/a	[52] [53] [54] [77]	I6YXT

NOTE: XHTML-Print defines a simple XHTML data stream suitable for printing. It is based on XHTML Basic.

Please note that both PrintEnhanced:1 and PrintBasic:1 do not have any mechanism to automatically delete the XHTML-Print document (on the +PR1+ or +PR2+) when it is no longer needed. It is recommended that printer control points monitor the progress of submitted jobs and remember to delete the XHTML file after the job is completed, aborted, or never consumed. One way to do this is to monitor the evented variable PE.JobEndState and the presence of the printer.

Please refer to the XHTML-Print/CSS Print Guidelines ([74]) for information on how to compose consistent XHTML.

Note that XHTML profile definitions define layout complexity and image profile definitions specify image complexity. The best way to avoid conflicts is for the XHTML profile to rely solely on the image profile definitions to define image limitations. (e.g. XHTML profiles are discouraged from imposing restrictions for things like pixel resolution or content binary lengths, unless already defined by an existing printing-related specification such as [77].

Note that 7.5.4.3.2.4.4 requires a DMPr to function correctly in scenarios where a DMS only supports a single HTTP connection.

Note that as long as the XHTML_PT profile can be supported, a Printing Device does not have to use the RANGE header to print the document. For example, if a Printer Device has enough buffer, it might print the XHTML_PT documents without the RANGE header. In other words, supporting the RANGE header is not required for a DMPr but supporting XHTML_PT profile is required.

7.4.2.3.5.2

[GUIDELINE] If a UPnP Printer's PrintEnhanced SCPD document indicates support for PNG, then the Printer shall be able to properly print the PNG images that are referenced in an XHTML-Print document that conforms to an XHTML profile that the Printer device supports.

Properly print and supports are as defined in guideline 7.4.2.3.5.1 above.

Support for PNG is defined in the PrintEnhanced state variable PE.XHTMLImageSupported. The value of this element shall be the DLNA-defined mime-type for PNG images (i.e. "image/png").

PNG images is defined as the set of permitted PNG image profiles defined in the XHTML profile definition.

[ATTRIBUTES]

M	A	DMPr	n/a	n/a	[53]	TA4M9	

7.4.2.3.5.3

[GUIDELINE] A UPnP Printer control point shall not send XHTML-Print documents containing references to PNG images unless the Printer indicates support for PNG as defined in the guideline above.

[ATTRIBUTES]

M	A	+PR1+ +PR2+	n/a	n/a	[53]	5I6YX	

7.4.2.3.5.4

[GUIDELINE] UPnP Printer control points shall follow these restrictions when creating XHTML-Print documents that are used with a UPnP Printer device.

- The XHTML-Print document used with a UPnP Printer device shall conform to a valid DLNA XHTML profile that is supported by the UPnP Printer device, as indicated by the printProfiles-AllRangeTrue-AllRangeFalse token.
- If the XHTML profile is XHTML_Baseline and the referenced image does not support the HTTP Range header, then the UPnP printer control point shall ensure that the content binary length of the image is equal to or less than 4194304 bytes (4 MB).
- Otherwise, the images referenced shall comply with the requirements of the XHTML profile used. (In this case, the size restrictions on the referenced images are determined by the image profile definitions.)

Used with a UPnP Printer device means that the Printer control point instructed the Printer device to print the XHTML-Print document and the images referenced in the document.

[ATTRIBUTES]

M	L	+PR1+ +PR2+	n/a	n/a	[53]	9TA4M	

NOTE: Since all XHTML profiles are organized hierarchically, an XHTML-Print document that conforms to a derived XHTML profile will automatically conform to the parent XHTML profile because derived profiles inherit all restrictions of parent profiles. For example, a document that conforms to XHTML_PT automatically conforms to XHTML_Complex.

Control points can acquire the file size in a variety of ways:

- by storing or serving the file directly,
- using res@size metadata,

- issuing an HTTP/1.0 HEAD request,
- by actually transfering the entire file to check its length, although this is not practical for large files.

7.4.2.3.5.5

[GUIDELINE] UPnP Printer control points shall not use Multi-part MIME encoding for the XHTML-Print document.

[ATTRIBUTES]

M	L	+PR1+ +PR2+	n/a	n/a	[37] [53]	2GRNX

NOTE: It is very difficult to guarantee that the printer will be able to print all of the included images, when encoded with Multi-part MIME.

PrintEnhanced:1 requires the printer to support Multi-part MIME but it is not recommended for applications in the future.

7.4.2.3.5.6

[GUIDELINE] A UPnP Printer control point shall be able to create and send XHTML-Print documents for the XHTML_Baseline profile.

[ATTRIBUTES]

M	A	+PR2+	n/a	n/a	[52] [53] [77]	YWKWY

NOTE: The use of XHTML_PT profile in the 3-Box Print System Usage requires the UPnP AV MediaServer to support the RANGE HTTP header on that content item. Since some UPnP AV MediaServers do not support the RANGE HTTP header for some content items, the use of the XHTML_PT profile is not always possible. In order to maintain minimum interoperability, the UPnP Printer control point needs to use the XHTML_Baseline profile for those content items.

7.4.2.3.5.7

[GUIDELINE] A UPnP Printer control point shall be able to create and send XHTML-Print documents for the XHTML_PT profile or XHTML_Baseline profile.

[ATTRIBUTES]

M	A	+PR1+	n/a	n/a	[52] [53] [77]	2WSTD

NOTE: In the 2-Box Print System Usage, the UPnP Printer control point always supports the RANGE HTTP header for all content items. The UPnP Printer control point is free to choose between the XHTML_PT and XHTML_Baseline profiles.

7.4.2.3.6 MM Photo Content selection

[GUIDELINE] UPnP Printer control points should allow the user to select the photos that they want to print and a choice of layouts.

[ATTRIBUTES]

S	A	+PR1+ +PR2+	n/a	n/a	[50] [88] [89] [90] [54]	GRNXA

NOTE: XHTML-Print photo templates are available for developer's convenience. Please refer to [54] for more information.

7.4.2.3.7 MM Printer selection

[GUIDELINE] UPnP Printer control points shall be capable of allowing the user to select the Printing Endpoint that they want to print to. This allows for the case when there is more then one image printer on the network.

[ATTRIBUTES]

M	A	+PR1+ +PR2+	n/a	n/a	[50] [88] [89] [90]	6X9Z5

7.4.2.3.8 MM Printer Status Monitoring

[GUIDELINE] UPnP Printer control points should monitor the status of the printer using the UPnP state variables and indicate to the user when there needs to be intervention (e.g. paper out).

[ATTRIBUTES]

S	A	+PR1+ +PR2+	n/a	n/a	[50] [88] [89] [90]	U6X9Z

7.4.2.3.9 MM Aquiring Image Content URIs

7.4.2.3.9.1

[GUIDELINE] UPnP Printer control points shall be able to insert the URIs for the image contents that exist locally into an XHTML-Print document.

[ATTRIBUTES]

M	R	+PR1+	n/a	n/a	[50] [88] [89] [90] [53]	LRK88

NOTE: These guideline indicates where image content URIs to be printed are located.

If a Device Class wants to support both the 2 box and 3 box Printing System Usages, it needs to support both the +PR1+ and +PR2+ Device Capabilities.

7.4.2.3.9.2

[GUIDELINE] UPnP Printer control points shall be able to insert the URIs for image contents on a UPnP AV MediaServer into an XHTML-Print document

[ATTRIBUTES]

M	R	+PR2+	n/a	n/a	[50] [88] [89] [90] [53]	RK883

7.4.2.3.10 MM PE:GetMediaList Response Syntax

7.4.2.3.10.1

[GUIDELINE] The syntax definition of the values of <MediaType> elements in *MediaList* output argument of PE:GetMediaList response shall be as follows:

- value-of-mediaType-element = mediaType-val *(mediaType-delim mediaType-val)
- mediaType-delim=SP
- mediaType-val=<the media type name as specified in [95] and defined as a vendor extension.>

Furthermore, <MediaType> element values shall not have preceding or trailing LWS characters.

[ATTRIBUTES]

M	L	DMPr	n/a	n/a	[90] [95]	883KP

NOTE: Due to the lack of the delimiter's definition for <MediaType> and <MediaSize> element's values in subclause 2.6.3.2 of [90], it seems that one or more whitespace(s), tab(s) or new line(s) can be used as delimiters for their values of their elements and those values are theoretically infinite in length. So, this guideline puts the reasonable limits for easy to implement devices and control points.

7.4.2.3.10.2

[GUIDELINE] The syntax definition of the values of <MediaSize> elements in *MediaList* output argument of PE:GetMediaList response shall be as follows:

- value-of-mediaSize-element = mediaSize-val *(mediaSize-delim mediaSize-val)
- mediaSize-delim=SP
- mediaSize-val=<the media size name as specified in [95] and defined as a vendor extension.>

Furthermore, <MediaSize> element values shall not have preceding or trailing LWS characters.

[ATTRIBUTES]

M	L	DMPr	n/a	n/a	[90] [95]	K883K

7.4.2.3.10.3

[GUIDELINE] If a UPnP Printer Control Point invokes a PE:GetMediaList action, it shall parse and interpret <MediaType> and/or <MediaSize> value(s) whose syntax are defined in 7.4.2.3.10.1 and 7.4.2.3.10.2 on a PE:GetMediaList response.

[ATTRIBUTES]

M	L	+PR1+, +PR2+	n/a	n/a	[90] [95]	9Z5YE

7.4.2.3.10.4

[GUIDELINE] If a UPnP Printer Control Point invokes a PE:GetMediaList action, it shall be tolerant of LWS characters as delimiters for <MediaType> and/or <MediaSize> value(s) on a PE:GetMediaList response. Furthermore it shall be tolerant of preceding or trailing LWS characters in <MediaType> and/or <MediaSize> value(s).

Tolerant behavior is defined as being able to successfully "parse and interpret" or "parse and ignore" values of those elements.

[ATTRIBUTES]

M	R	+PR1+ +PR2+	n/a	n/a	[90] [95]	RNXAZ

NOTE: This guideline ensures that a UPnP Printer Control Point will continue to behave properly even if a UPnP Printer device, non-DLNA compliant device, returns PE:GetMediaList response with the incompatible syntax defined in 7.4.2.3.10.1 and 7.4.2.3.10.2.

Vendors are free to implement a UPnP Printer Control Point which can parse and interpret linear white space characters (LWS) on <MediaType> and/or <MediaSize> values.

7.4.2.3.11 MM Printer Media Guidelines

7.4.2.3.11.1

[GUIDELINE] A UPnP Printer device shall inform the list of its supported media via the SCPD and the PE:GetMediaList(none, none).

Informing the list of its supported media via SCPD is defined in 7.4.2.3.11.2.

Informing the list of its supported media via PE:GetMediaList(none, none) is defined in 7.4.2.3.11.3.

In subclause 7.4.2.3.11, PE:GetMediaList(a, b) means the action request that has "a" value for its MediaSize argument and "b" value for its MediaType argument.

[ATTRIBUTES]

M	R	DMPr	n/a	n/a	[90]	X9Z5Y	

7.4.2.3.11.2

[GUIDELINE] The list of media described as the allowed value list for the MediaSize and the MediaType state variable in the SCPD shall be the static list which includes all possible media at the device. It shall also include "none" and "device-setting" as required in [90].

[ATTRIBUTES]

M	C	DMPr	n/a	n/a	[90]	A4M93	

NOTE: For consumer printers, the size and type of printing papers can be changed just before starting to print. So, UPnP Printer devices can only use the subset of all possible media at boot. However, this guideline clarifies that UPnP Printer devices need to expose all the possible media in order for a UPnP Printer Control Point to know all the possible media via SCPD.

7.4.2.3.11.3

[GUIDELINE] The list of media reported via the PE:GetMediaList(none, none) shall be either

- the dynamic list which includes only the available media at this point, or
- the static list which includes all possible media at the device.

[ATTRIBUTES]

M	C	DMPr	n/a	n/a	[90]	NXAZ2	

NOTE: It is assumed that a user selects a proper media at the printing time.

7.4.2.3.11.4

[GUIDELINE] A UPnP Printer device shall inform the current default media size information via PE:GetMediaList(device-setting, none). The reported information shall include media size information. The "none", "device-setting" or empty string shall not be returned as the default size information.

[ATTRIBUTES]

M	R	DMPr	n/a	n/a	[90]	4M93F	

7.4.2.3.11.5

[GUIDELINE] A UPnP Printer device shall inform the current default media type information via PE:GetMediaList(none, device-setting). The reported information shall include media type information. The "none", "device-setting" or empty string shall not be returned as the default type information.

[ATTRIBUTES]

M	R	DMPr	n/a	n/a	[90]	YXTQP	

7.4.2.3.11.6

[GUIDELINE] A UPnP Printer device may support any media.

[ATTRIBUTES]

O	R	DMPr	n/a	n/a	[90]	9K78I

NOTE: DLNA does not define required media for printing.

7.4.2.3.11.7

[GUIDELINE] A UPnP Printer device shall use the media size name conforming to [95].

[ATTRIBUTES]

M	R	DMPr	n/a	n/a	[95]	6YXTQ

NOTE: Media size name is used to identify the media size in PrintEnhanced:1.

7.4.2.3.11.8

[GUIDELINE] For media sizes defined in [95], a UPnP Printer device shall specify the entire media size name, including the short dim and long dim part of the media size name as defined in [95]. When the printer reports the media names that it supports it shall use the appropriate units of measure for the media. Using mm's when the paper size is commonly measured in inches (or vice versa) is prohibited.

[ATTRIBUTES]

M	L	DMPr	n/a	n/a	[90] [95]	K78IW

NOTE: Dimension part is the last part of the media size name as defined in [95]

Table 31 defines dimension parts for some papers not defined in [95].

For example, media size names used in the Photo Template specification [54] have following dimension parts.

- Letter: 8.5x11in
- 4" x 6": 4x6in
- A4: 210x297mm
- A6: 105x148mm
- Hagaki: 100x148mm
- L size paper: 89x127mm

7.4.2.3.11.9

[GUIDELINE] For media sizes listed in Table 31, a UPnP Printer device shall use the dimension part of the media size name as defined in Table 31.

Table 31 — Media Size Dimensions

Common Name	Media Size Dimension
L Size Paper (89 x 127 mm)	89x127mm
2L Size Paper (127 x 178mm)	127x178mm
HV Size Paper (89 x 158mm)	89x158mm
Hi-Vision Size Paper (102 x 181mm)	101.6x180.6mm
Credit Card Size Paper (54 x 86mm)	54x86mm
Card Size Paper (55x91mm)	55x91mm

[ATTRIBUTES]

M	L	DMPr	n/a	n/a	[90] [95]	C5BHC	

7.4.2.3.11.10

[GUIDELINE] For media sizes defined in neither [95] nor Table 31, a UPnP Printer device may use the vendor dependent dimension part as long as the syntax conforms to [95].

[ATTRIBUTES]

O	R	DMPr	n/a	n/a	[95]	X7W7Y	

7.4.2.3.11.11

[GUIDELINE] A UPnP Printer control point shall identify media size via only the dimension part of the media size name.

[ATTRIBUTES]

M	L	+PR1+ +PR2+	n/a	n/a	[90] [95]	E52V7	

7.4.2.3.11.12

[GUIDELINE] A UPnP Printer control point shall use media size names retrieved from the UPnP Printer device for the input argument to specify media size name, e.g. the MediaSize argument of the PE:CreateURIJob action.

[ATTRIBUTES]

M	C	+PR1+ +PR2+	n/a	n/a	[90] [95]	52V7P	

7.4.2.3.11.13

[GUIDELINE] A UPnP Printer device shall use the same dimension part of the media size names for both portrait and landscape media.

[ATTRIBUTES]

M	R	DMPr	n/a	n/a	[95]	GH9DD	

NOTE: [95] defines the dimension part and it needs to be in the following form, "short-dim" "x" "long-dim".

For example, the dimension part of the A4 landscape media name needs to be 210x297mm, not 297x210mm

7.4.2.3.12 MM Printer Margining Guidelines

7.4.2.3.12.1

[GUIDELINE] A UPnP Printer device shall inform its margin information for each of the four sides via PE:GetMargins action.

[ATTRIBUTES]

M	R	DMPr	n/a	n/a	[90]	5BHC8	

NOTE: PE:GetMargins action is used to indicate non-printable area.

7.4.2.3.12.2

[GUIDELINE] A UPnP Printer device shall not set FullBleedSupported out parameter to '1' if it does not support zero-margin printing.

Zero-margin printing means printing with specifying all-zero margins.

[ATTRIBUTES]

M	C	DMPr	n/a	n/a	[90]	7W7YP	

NOTE: Note that returning non-zero margins with FullBleedSupported '1' is legal as defined in [90].

Many printers can easily print borderless photos by slightly enlarging the photo and over printing the edge of the paper. Excess ink is captured in a gutter. However, it is extremely difficult to accurately print to a very small margin, for example 0.5mm, due to mechanical tolerances. This is why there are two mechanisms for borderless. For example, a typical printer might report 3mm margins and "true" for the borderless boolean. It is telling the Control Point not to specify a margin between 0 and 3mm.

7.4.2.3.12.3

[GUIDELINE] A UPnP Printer device shall set any value, i.e, '0' or '1' to the FullBleedSupported out parameter if it sets the PageMargins out parameter to all-zero.

[ATTRIBUTES]

M	R	DMPr	n/a	n/a	[90]	93RE5	

NOTE: In case that the PageMargins is all-zero, zero-margin printing is available regardless of the FullBleedSupported value.

7.4.2.3.12.4

[GUIDELINE] A UPnP Printer Control Point shall not issue zero-margin printing unless the PageMargins out parameter has a value of all-zero or the FullBleedSupported out parameter has a value of '1'.

[ATTRIBUTES]

M	C	+PR1+ +PR2+	n/a	n/a	[90]	ZFPWW	

7.4.2.3.12.5

[GUIDELINE] A UPnP Printer Control Point shall specify the allowed margins into the CSS based on the information obtained via PE:GetMargins when it does not issue zero-margin printing.

As repeated in 10.3.3.12 in [56], margins may be specified in the several forms in the CSS/XHTML document.

The allowed margins above means that in any forms, for each of the four sides, the specified margin shall be equal to or larger than the margin the printer reports as supported.

[ATTRIBUTES]

M	C	+PR1+ +PR2+	n/a	n/a	[90] [56]	3RE5V	

7.4.3 Content Synchronization MM/CM Guidelines

7.4.3.1 General

The guidelines contained in 7.4.3 apply only when the Upload or Download Synchronization Controller System Usages are implemented by a UPnP AV MediaServer control point and a UPnP AV MediaServer.

7.4.3.2 MM/CM: Download Synchronization Controller

7.4.3.2.1

[General] The Content Synchronization MM/CM Guidelines subclause defines the DLNA Upload and Download Synchronization Controller System Usage requirements for a UPnP AV MediaServer and a Synchronization Controller Device Capability. The following conditionally mandatory requirements are adhered to only when supporting the Upload or Download Synchronization System Usages.

7.4.3.2.2

[Guideline] A Download Synchronization Controller shall implement a UPnP AV MediaServer control point for synchronizing content from a DMS or M-DMS respectively.

[Attributes]

M	A	+DNSYNC+	n/a	n/a	[46] [49]	73OLY

NOTE: This guideline indicates that a Download Synchronization Controller Device Capability will use a UPnP control point that controls a UPnP AV MediaServer for browsing, searching and synchronizing content from the Media Server.

7.4.3.2.3

[Guideline] A Download Synchronization Controller shall implement a UPnP AV MediaServer control point capable of invoking the following actions:

- CDS:Search
- CDS:Browse

[Attributes]

M	A	+DNSYNC+	n/a	n/a	[46] [49]	V8WTG

NOTE: This guideline clarifies the UPnP CDS actions a Download Synchronization Controller needs to be able to invoke with a UPnP AV MediaServer.

7.4.3.2.4

[Guideline] A Download Synchronization Controller may maintain synchronization of some or all of the metadata, items, and resources exposed by a DMS/M-DMS implementing Content Synchronization Device Option.

[Attributes]

O	C	+DNSYNC+	n/a	n/a	[46] [49]	BVYZW

NOTE: This guideline clarifies that it is the role of the Download Synchronization Controller to decide whether some or all of the items, metadata, and resources need to be tracked to constitute a completed synchronization.

7.4.3.3 MM/CM: Upload Synchronization Controller

7.4.3.3.1

[General] The Content Synchronization MM/CM Guidelines subclause defines the DLNA Upload and Download Synchronization Controller System Usage requirements for a UPnP AV MediaServer and a Synchronization Controller Device Capability. The following conditionally mandatory requirements are adhered to only when supporting the Upload or Download Synchronization System Usages.

7.4.3.3.2

[Guideline] An Upload Synchronization Controller shall implement a UPnP AV MediaServer control point for synchronizing content to a DMS or M-DMS respectively.

[Attributes]

M	A	+UPSYNC+	n/a	n/a	[46] [49]	3OLY3	

NOTE: This guideline indicates that an Upload Synchronization Controller Device Capability will use a UPnP control point that controls a UPnP AV MediaServer for sending content to the MediaServer.

7.4.3.3.3

[Guideline] An Upload Synchronization Controller shall implement a UPnP AV MediaServer control point capable of invoking the following actions.

- CDS:CreateObject.
- CDS:UpdateObject
- CDS:DestroyObject
- CDS:GetFeatureList
- CDS:Search
- CDS:Browse
- CDS:GetServiceResetToken

[Attributes]

M	L	+UPSYNC+	n/a	n/a	[46] [49]	MURA8	

NOTE: The capability of invoking these actions is required for performing the following optional content management operations: OCM: upload content, OCM: create child container, OCM: destroy object and OCM: change metadata.

7.4.3.3.4

[Guideline] An Upload Synchronization Controller shall be capable of invoking the OCM: upload content operation.

See 7.4.1.7.2 for more information about optional content management operations.

[Attributes]

M	L	+UPSYNC+	n/a	n/a	[46] [49]	8WTGH	

NOTE: Allows an Upload Synchronization Controller to synchronize (add) new content to an AV MediaServer.

7.4.3.3.5

[Guideline] An Upload Synchronization Controller shall be capable of invoking the OCM: create child Container operation.

See 7.4.1.7.2 for more information about optional content management operations.

[ATTRIBUTES]

M	L	+UPSYNC+	n/a	n/a	[46] [49]	VYZW3	

NOTE: Allows an Upload Synchronization Controller to invoke a necessary synchronization (adding new containers) with an AV MediaServer.

7.4.3.3.6

[GUIDELINE] An Upload Synchronization Controller shall be capable of invoking the OCM: destroy object Operation.

See 7.4.1.7.2 for more information about optional content management operations.

[ATTRIBUTES]

M	L	+UPSYNC+	n/a	n/a	[46] [49]	URA8J	

NOTE: Allows an Upload Synchronization Controller to invoke a necessary synchronization (removing non-restricted content or containers) from an AV Media Server.

7.4.3.3.7

[GUIDELINE] An Upload Synchronization Controller shall be capable of invoking the OCM: change metadata operation.

See 7.4.1.7.2 for more information about optional content management operations.

[ATTRIBUTES]

M	L	+UPSYNC+	n/a	n/a	[46] [49]	OLY3A	

NOTE: Allows and Upload Synchronization Controller to invoke a necessary synchronization (modifying content) on an AV MediaServer.

7.4.3.3.8

[GUIDELINE] An Upload Synchronization Controller may upload some or all of the optional items, metadata, and resources it decides to track on a DMS/M-DMS implementing the Content Synchronization Device Option.

[ATTRIBUTES]

O	C	+UPSYNC+	n/a	n/a	[46] [49]	WTGHW	

NOTE: This guideline clarifies that it is the role of the Upload Synchronization Controller to decide what optional items, metadata and resources need to be uploaded and tracked to constitute a completed synchronization.

7.4.3.4 MM/CM General Rules for Thrashing Avoidance

7.4.3.4.1

[GENERAL] The Content Synchronization MM/CM Guidelines subclause defines the DLNA Upload and Download Synchronization Controller System Usage requirements for a UPnP AV MediaServer and a Synchronization Controller Device Capability. The conditionally mandatory requirements in this subclause are adhered to only when supporting the Upload or Download Synchronization System Usages.

7.4.3.4.2

[GUIDELINE] An Upload Synchronization Controller shall not propagate without external input the local stored metadata or content to the Media Server for objects that have changed on the server but not on the client since the last synchronization.

[ATTRIBUTES]

M	A	+UPSYNC+	n/a	n/a	[46] [49]	YZW3Y	

NOTE: Limits the number of times competing Synchronization Controllers can change content on an AV Media Server without external input, such as user intervention.

7.4.3.5 MM/CM: DMS or M-DMS with Content Synchronization Device Option Support Definition

7.4.3.5.1

[GENERAL] The Content Synchronization MM/CM Guidelines subclause defines the DLNA Upload and Download Synchronization Controller System Usage requirements for a UPnP AV MediaServer and a Synchronization Controller Device Capability. The conditionally mandatory requirements in this subclause are adhered to only when supporting the Upload or Download Synchronization System Usages.

7.4.3.5.2

[GUIDELINE] A UPnP AV MediaServer shall advertise the Content Synchronization Device Option by specifying the <dlna:X_DLNACAP> element (as a child of the <device> element that represents the MediaServer) with the *content-synchronization token*.

Table 32 — Capability ID Syntax

Capability ID	Description
content-synchronization	The UPnP AV MediaServer supports the DLNA Content Synchronization Device option.
The "content-synchronization" portion of the capability ID is a literal, string value.	

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46] [49]	LY3AY	

NOTE: An Upload/Download synchronization controller uses this method to determine if it can synchronize with a particular DMS/M-DMS.

7.4.3.5.3

[GUIDELINE] A UPnP AV MediaServer shall implement the following optional content management operations:

- OCM: update content
- OCM: create child container
- OCM: destroy object
- OCM: change metadata

as defined in 7.4.1.7.2.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	RA8JY	

NOTE: These OCM operations allow content to be synchronized between an AV MediaServer and the +UPSYNC+ or +DNSYNC+ client capabilities.

7.4.3.5.4

[GUIDELINE] A UPnP AV MediaServer shall implement the Tracking Changes Option of the ContentDirectory service.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	TGHWY	

NOTE: As well as changes that occur as the result of UPnP Actions when the Content Directory service is in the on-line state, changes to the underlying schema can occur through other mechanisms both in the online and off-line state. If, for example, a user deletes files manually from the schema that the Content Directory service is representing while it is in the off-line state, these changes will be visible to Control Points when the Content Directory Service comes back on line. Similarly, if a user deletes files manually from the schema when the Content Directory Service is on-line, these changes will be handled as if the changes occurred via UPnP Actions.

7.4.3.5.5

[GUIDELINE] A UPnP AV MediaServer shall implement the LastChange evented state variable.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46] [49]	ZW3YH	

NOTE: The LastChange state variable allows the accumulation (buffering) and delivery (eventing) of all changes to an AV MediaServer's content. The LastChange state variable is a required feature of the Tracking Changes Option of the ContentDirectory service.

7.4.3.5.6

[GUIDELINE] A UPnP AV MediaServer shall form a LastChange state variable consisting of a properly formed XML document as per [96].

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46] [49] [96]	Y3AY5	

7.4.3.5.7

[GUIDELINE] A UPnP AV MediaServer shall send the LastChange event message within 10 seconds of a change occurring on the CDS.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	A8JYO	

7.4.3.5.8

[GUIDELINE] A UPnP AV MediaServer shall implement the CDS:Search() operation.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46] [49]	GHWYR	

NOTE: CDS:Search() is the mechanism for discovering what has changed that the CDS is tracking and is a required feature of the Tracking Changes Option of the ContentDirectory service.

7.4.3.5.9

[GUIDELINE] All UPnP AV MediaServer CDS container objects shall include the @searchable attribute with a value of "1".

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	W3YHF

NOTE: All containers on the CDS needs to be searchable. A search can start from any container in the CDS.

7.4.3.5.10

[GUIDELINE] A UPnP AV Media Server shall list at least the following metadata items in the SearchCapabilities state variable.

- upnp:containerUpdateID
- upnp:objectUpdateID
- upnp:class
- @parentID
- @id

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	3AY53

NOTE: These values are the minimal set to allow searching for changes.

7.4.3.5.11

[GUIDELINE] A UPnP AV Media Server shall list at least the following metadata items in the SortCapabilities state variable.

- upnp:containerUpdateID
- upnp:objectUpdateID

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	8JYO5

NOTE: These allow ordering of changes in the CDS:Search() results.

7.4.3.5.12

[GUIDELINE] A UPnP AV MediaServer shall implement at least the following operators on the indicated metadata items within the SearchCriteria of the CDS:Search() operation.

Table 33 — UPnP AV MediaServer Metadata SearchCriteria

Metadata Item	Operator(s)
upnp:containerUpdateID	>,=,>=,<,<=,!=, exists
upnp:objectUpdateID	>,=,>=,<,<=,!=, exists
upnp:class	=,derivedFrom
parentID	=
id	=

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	HWYRV	

7.4.3.5.13

[GUIDELINE] A UPnP AV Media Server shall implement in the SearchCriteria argument of the CDS:Search() action the logical 'and' and 'or' operators on at least 5 clauses containing the metadata and operators specified in 7.4.3.5.12.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	3YHFF	

7.4.3.5.14

[GUIDELINE] A UPnP AV MediaServer shall implement AV CDS Tracking Changes Option for all items and therefore the following metadata attributes for all CDS items.

- upnp:objectUpdateID
- upnp:res@updateCount

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	AY533	

NOTE: These metadata items allow tracking of CDS content entries. There is no partial support of content tracking for some content on the server and not for other content.

7.4.3.5.15

[GUIDELINE] A UPnP AV MediaServer shall implement AV CDS Tracking Changes Option for all containers and therefore the following metadata attributes for all CDS containers.

- upnp:objectUpdateID
- upnp:totalDeletedChildCount
- upnp:containerUpdateID
- upnp:childCount

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	JYO5N	

NOTE: These metadata items allow tracking of CDS container level changes. There is no partial support of content tracking for some content on the server and not for other content.

7.4.3.6 MM/CM: Support for res@dlna:estimatedSize

7.4.3.6.1

[GENERAL] The Content Synchronization MM/CM Guidelines subclause defines the DLNA Upload and Download Synchronization Controller System Usage requirements for a UPnP AV MediaServer and a Synchronization Controller Device Capability. The conditionally mandatory requirements in this subclause are adhered to only when supporting the Upload or Download Synchronization System Usages.

The res@dlna:estimatedSize is used by the server to advertise an estimate of the size of a content binary in those cases where the exact size is not known. For example, if the content requires transcoding that will not be performed until a request for the content is received, the exact length of the transcoded version cannot be known. However, it can be useful for downloading or synchronizing endpoints to have an estimate of the size of a content binary to determine whether it is generally possible for that content to fit within the available storage.

7.4.3.6.2

[Guideline] A UPnP AV Media Server may include the res@dlna:estimatedSize attribute for any resource within a content object returned as part of the result of a CDS:Browse() or CDS:Search() operation.

[Attributes]

O	A	DMS	M-DMS	n/a	[46] [49]	WYRVV

7.4.3.6.3

[Guideline] A UPnP AV Media Server Upload Controller may include the res@dlna:estimatedSize attribute for any resource created as part of the CDS:CreateObject() call within an OCM: create object operation.

[Attributes]

O	A	+UP+	n/a	n/a	[46] [49]	YHFF8

7.4.3.6.4

[Guideline] A UPnP AV Media Server Upload Synchronization Controller may include the res@dlna:estimatedSize attribute for any resource created as part of the CDS:CreateObject() call within an OCM: create object operation.

[Attributes]

O	A	+UPSYNC+	n/a	n/a	[46] [49]	Y5337

7.4.3.6.5

[Guideline] If present the res@dlna:estimatedSize attribute shall have the same syntax and type as the res@size attribute.

[Attributes]

M	A	DMS +UP+ +UPSYNC+	M-DMS	n/a	[46] [49]	YO5NQ

7.4.3.6.6

[Guideline] If the res@Size attribute is present for any resource then the res@estimatedSize attribute shall not be present for the same resource.

[Attributes]

M	A	DMS +UP+ +UPSYNC+	M-DMS	n/a	[46] [49]	YRVVM

7.4.3.7 MM/CM: Operations that need CDS:UpdateObject

7.4.3.7.1

[General] The Content Synchronization MM/CM Guidelines subclause defines the DLNA Upload and Download Synchronization Controller System Usage requirements for a UPnP AV MediaServer and a Synchronization Controller Device Capability. The conditionally mandatory requirements in this subclause are adhered to only when supporting the Upload or Download Synchronization System Usages.

7.4.3.7.2

[Guideline] A UPnP AV MediaServer shall implement the OCM: change metadata operation and therefore shall implement CDS:UpdateObject action.

[Attributes]

M	A	DMS	M-DMS	n/a	[46] [49]	HFF8I

NOTE: The CDS:UpdateObject action is used to change metadata on an existing CDS object.

7.4.3.8 MM/CM: General Rules for CDS:UpdateObject Request Syntax

7.4.3.8.1

[General] The Content Synchronization MM/CM Guidelines subclause defines the DLNA Upload and Download Synchronization Controller System Usage requirements for a UPnP AV MediaServer and a Synchronization Controller Device Capability. The conditionally mandatory requirements in this subclause are adhered to only when supporting the Upload or Download Synchronization System Usages.

7.4.3.8.2

[Guideline] A UPnP AV MediaServer control point invoking CDS:UpdateObject shall apply the mandatory metadata encoding rules for the following guidelines in the NewTagValue input argument.

- 7.4.1.3.1 MM UPnP AV Control Point Tolerance of Unknown Property
- 7.4.1.3.2 MM DIDL-Lite Restrictions
- 7.4.1.3.3 MM DIDL-Lite Max Metadata Length
- 7.4.1.3.5 MM DIDL-Lite Boolean Values
- 7.4.1.3.6 MM upnp:class Values
- 7.4.1.3.7 MM DIDL-Lite dc:date Format
- 7.4.1.3.9 MM DIDL-Lite Desc Element Use
- 7.4.1.3.10 MM URI Rules
- 7.4.1.3.11 MM DIDL-Lite Recommended Metadata Properties
- 7.4.1.3.12 MM protocolInfo Context
- 7.4.1.3.15 MM DIDL-Lite protocolInfo values
- 7.4.1.3.16 MM protocolInfo values: 4th Field
- 7.4.1.3.17 MM pn-param (DLNA.ORG_PN Parameter)
- 7.4.1.3.22 MM ci-param (Conversion Indicator Flag)

[Attributes]

M	A	+UPSYNC+	n/a	n/a	[46] [49]	5337Q

7.4.3.8.3

[Guideline] A UPnP MediaServer Control Point invoking CDS:UpdateObject shall not include the

- @dlna:dlnaManaged
- res@dlna:ifoFileURI
- res@dlna:importIfoFileURI

- albumArtURI@dlna:profileID
- res@dlna:trackTotal
- res@dlna:resumeUpload
- @dlna:containerType
- res@dlna:uploadedSize

metadata properties in the NewTagValue input argument.

[Attributes]

M	A	+UPSYNC+	n/a	n/a	[46] [49]	O5NQX	

NOTE: These values are set by the MediaServer capabilities and are defined as read-only attributes.

7.4.3.8.4

[Guideline] A UPnP AV MediaServer control point invoking CDS:UpdateObject shall apply the rules specified in 7.4.1.3.4.1 and 7.4.1.3.4.2 to all metadata.

[Attributes]

M	A	+UPSYNC+	n/a	n/a	[46] [49]	RVVM7	

7.4.3.8.5

[Guideline] A UPnP AV MediaServer control point invoking CDS:UpdateObject, should not delete metadata properties as defined in guideline 7.4.1.3.11 for the given upnp:class.

[Attributes]

S	A	+UPSYNC+	n/a	n/a	[46] [49]	FF8IA	

NOTE: The Control Point will preferably not delete metadata recommended for a given upnp:class.

7.4.3.8.6

[Guideline] A UPnP AV MediaServer control point invoking CDS:UpdateObject with optional metadata shall properly declare all namespaces.

[Attributes]

M	A	+UPSYNC+	n/a	n/a	[46] [49][49]	337QN	

7.4.3.8.7

[Guideline] A UPnP AV MediaServer allowing an update to the <dc:date> elements shall allow the MediaServer control point to specify any valid form of <dc:date>, as defined by guideline 7.4.1.3.7.

[Attributes]

M	A	DMS	M-DMS	n/a	[46] [49]	5NQX4	

NOTE: This guideline allows the control point to specify any valid form for <dc:date>. Control points are not to assume that the format they specified in the request will be the final form of the <dc:date>.

7.4.3.8.8

[Guideline] A UPnP AV MediaServer implementing the OCM: change metadata operation shall ignore attempts to add unsupported metadata and only generate an error response if some other error occurs or there were no successful modifications to the target object.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	VVM7J	

NOTE: The UPnP CDS behavior allows error conditions to be generated when attempting to add unsupported metadata. Since the CP does not currently have a reliable method for determining which metadata is actually supported by the DMS/M-DMS, restricting this error response allows supported metadata to be accepted while simultaneously allowing unsupported metadata to have no effect.

7.4.3.9 MM/CM: General Rules for Server Behavior for CDS:UpdateObject

7.4.3.9.1

[GENERAL] The Content Synchronization MM/CM Guidelines subclause defines the DLNA Upload and Download Synchronization Controller System Usage requirements for a UPnP AV MediaServer and a Synchronization Controller Device Capability. The conditionally mandatory requirements in this subclause are adhered to only when supporting the Upload or Download Synchronization System Usages.

7.4.3.9.2

[GUIDELINE] A UPnP AV MediaServer implementing the OCM: change metadata operation for a particular object should accept the creation of recommended properties corresponding to the specified upnp:class, which are defined in the guideline 7.3.25 MM DIDL-Lite Recommended Metadata Properties.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46] [49]	F8IAT	

7.4.3.9.3

[GUIDELINE] A UPnP AV MediaServer implementing the OCM: change metadata operation for a particular object shall be able to accept CDS:UpdateObject requests that specify changes to multiple properties.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	37QN7	

7.4.3.9.4

[GUIDELINE] A UPnP AV MediaServer implementing the OCM: change metadata operation for a particular object and receiving a request where the CurrentTagValue does not match the current state of the CDS, shall respond with an error code 702 (Invalid currentTagValue).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	NQX4R	

7.4.3.10 MM/CM: OCM: Change Metadata Operation

7.4.3.10.1

[GENERAL] The Content Synchronization MM/CM Guidelines subclause defines the DLNA Upload and Download Synchronization Controller System Usage requirements for a UPnP AV MediaServer and a Synchronization Controller Device Capability. The conditionally mandatory requirements in this subclause are adhered to only when supporting the Upload or Download Synchronization System Usages.

7.4.3.10.2

[GUIDELINE] A UPnP AV Media Server Control Point invoking an OCM: change metadata operation, shall invoke CDS:UpdateObject with the following rules.

- The ObjectID shall identify the CDS object that is to receive the metadata change and the CDS object shall support the OCM: change metadata operation..

[ATTRIBUTES]

M	A	+UPSYNC+	n/a	n/a	[46] [49]	7GY63	

7.4.3.10.3

[GUIDELINE] A UPnP AV MediaServer implementing the OCM: change metadata operation, shall ignore all <res> elements received in a NewTagValue argument of the CDS:UpdateObject request.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	84T7U	

NOTE: The resource data defines the nature of the content and cannot be changed without also changing the binary data. Creating a new <res> element is not supported, if the control point desires to upload content, it nereds to create a new object with the Upload AnyContainer or OCM: upload content operations. Additionally, the OCM: change metadata operation is not used for changing the binary data of a <res> element, the object will be deleted and re-created.

7.4.4 Scheduled Recording Media Management Guidelines

7.4.4.1 MM/SR System Usage Feature Support

7.4.4.1.1

[GENERAL] 7.4.4 defines the DLNA Scheduled Recording System Usage requirements for a UPnP AV MediaServer and a Scheduled Recording Controller Device Capability. The guidelines in 7.4.4 are mandatory if the Scheduled Recording System Usage is implemented.

7.4.4.1.2

[GUIDELINE] If a DMS or M-DMS contains a UPnP ScheduedRecording service in the UPnP AV MediaServer device, then it shall conform to all of the guidelines for a UPnP AV MediaServer in 7.4.4 as defined in 7.4.4.1 through 7.4.4.30 inclusive.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[49] [60]	59OVP	N

NOTE: This guideline classifies the group of guideline requirements that a DMS and M-DMS implements to support the Scheduled Recording Device Option for the Scheduled Recording System Usage

7.4.4.1.3

[GUIDELINE] A DLNA device class may implement the Scheduled Recording Controller Device Capability.

[ATTRIBUTES]

O	A	DMS DMP DMR DMC DMPr	M-DMS M-DMP M-DMC M-DMD M-DMU	n/a	[46] [49] [60]	AIOZ5	N

7.4.4.1.4

[GUIDELINE] A Scheduled Recording Controller shall implement a UPnP AV MediaServer control point that interacts with the ContentDirectory service for browsing content and with the ScheduledRecording service for creating and managing recordings. It shall also conform to all of the guidelines for a Scheduled Recording Controller in 7.4.4 as defined in 7.4.4.1 through 7.4.4.30 inclusive.

[ATTRIBUTES]

M	A	+SR+	n/a	n/a	[46] [49] [60]	D7M8R	N

NOTE: This guideline classifies the group of guideline requirements that a Scheduled Recording Controller implements to support the Scheduled Recording System Usage.

7.4.4.2 MM/SR Exposing Recorded Content

7.4.4.2.1

[GUIDELINE] A UPnP AV MediaServer should expose its recorded content in a ContentDirectory service for DLNA consumption.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46] [49]	K7CVW	N

NOTE: Recording devices are ultimately built to offer consumption of their recorded content. There are cases where the content consumption method is not defined by DLNA System Usages.

7.4.4.2.2

[GUIDELINE] If a UPnP AV MediaServer exposes recorded content in a ContentDirectory service, then the recorded content shall be exposed in the ContentDirectory service co-located with the ScheduledRecording service.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	OATA6	N

7.4.4.2.3

[GUIDELINE] If a UPnP AV MediaServer exposes recorded content in a ContentDirectory service, then at least one res property of the recorded content shall be conformant to a DLNA Media Format Profile as defined in 7.4.1.3.15.4.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	WZAIZ	N

7.4.4.2.4

[GUIDELINE] If a UPnP AV MediaServer supports simultaneous streaming and recording, then the recorded content CDS object with a res property should exist in the CDS as soon as the recordTask starts recording.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46] [49]	55MW8	N

7.4.4.2.5

[GUIDELINE] If a UPnP AV MediaServer supports simultaneous streaming and recording, then the recorded content CDS object should follow the DLNA UCDAM buffer model guidelines 7.4.1.3.32, 7.4.1.3.33 and Annex E to allow a UPnP AV MediaServer control point to interact with the dynamic content binary size.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46] [49]	ZR24T	N

7.4.4.2.6

[GUIDELINE] A UPnP AV MediaServer shall expose the arib:objectType property in the Japan region or dlna:objectType property in other geographical regions in the recorded content CDS item with a value of the applicable broadcast system, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.

The namespace "urn:schemas-arib-or-jp:elements-1-0/" shall be specified in the <item> element or the <arib:objectType> element and the namespace prefix shall be "arib" when exposing the arib:objectType property.

The namespace "urn:schemas-dlna-org:device-1-0" shall be specified in the <item> element or the <dlna:objectType> element and the namespace prefix shall be "dlna" when exposing the dlna:objectType property.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	9M53Q	N

NOTE: This requirement provides UPnP AV MediaServer implementations with a CDS property which can be used by control points to identify content items that were created as a result of an execution of a recordTask. Values for the arib:objectType property need to be as specified in [104] and [105]. Values for the dlna:objectType will need to something meaningful for the applicable broadcast systems. The following table provides some initial guidance that can be updated and added to the future:

Broadcast Systems	dlna:objectType Values
US Terrestrial Broadcast Systerm	ATSC_TB
US Cable Broadcast Systems	TBD
European Broadcast Systems	TBD

Table 34 — dlna:objectType Values

7.4.4.2.7

[GUIDELINE] A UPnP AV MediaServer shall expose the dc:date property in the recorded content CDS item with a non-empty and non-whitespace value, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The CDS item of the recorded content exposes a upnp:class property with a value of object.item.audioItem or object.item.videoItem.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	LRZI6	N

7.4.4.2.8

[GUIDELINE] A UPnP AV MediaServer shall set the value of the dc:date property in the recorded content CDS item with a value which indicates the actual start date and time of the recorded content, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The CDS item of the recorded content exposes the dc:date property.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	HK2ET	N

NOTE: This is the CDS property that is recommended for containing the actual date and time of the recorded content instead of using the upnp:recordedStartDateTime property. Usage of this property makes this consistent for all types of CDS items contained in the CDS.

7.4.4.2.9

[GUIDELINE] A UPnP AV MediaServer shall set the value of the res@duration property in the recorded content CDS item with a value which indicates the recorded duration of the recorded content after the completion of the recording, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The CDS item of the recorded content exposes the dc:duration property.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	HMUQV	N

NOTE: This is the CDS property that is recommended for containing the actual duration of the recorded content instead of using the upnp:recordedDuration property. Usage of this property makes this consistent for all types of CDS items contained in the CDS.

7.4.4.2.10

[GUIDELINE] A UPnP AV MediaServer shall conform to the guidelines listed in Table 35 when all the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The CDS srs:class property value is from the recordSchedule object that spawned the recordTask.

srs:class Value	Guidelines for Recorded CDS Properties
OBJECT.RECORDSCHEDULE.DIRECT.CDSNONEPG	7.4.4.9.5 – 7.4.4.9.11
OBJECT.RECORDSCHEDULE.DIRECT.MANUAL	7.4.4.10.5 – 7.4.4.10.7
OBJECT.RECORDSCHEDULE.DIRECT.CDSEPG	7.4.4.11.6 – 7.4.4.11.12
OBJECT.RECORDSCHEDULE.QUERY.CONTENTNAME	7.4.4.12.5
OBJECT.RECORDSCHEDULE.QUERY.CONTENTID	7.4.4.13.5

Table 35 — Guidelines for Recorded CDS Properties based on srs:class values

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	SXEVC	N

NOTE: Depending on the record class used for schedule a recording, these conformance guidelines provide the recommended CDS property values for the recorded content CDS item. Note that the same CDS properties defined for each class are semantically equivalent, but guidelines are duplicated for each applicable record class (e.g. upnp:channelID). The following table summarizes the recommended CDS properties for the recorded content for each srs:class:

srs:class Value	Recommended Recorded CDS Properties
OBJECT.RECORDSCHEDULE.DIRECT.CDSNONEPG	res@dlna:scheduledRecordedContent, dc:date, res@duration, dc:title, upnp:channelName, upnp:channelNr, upnp:genre, upnp:channelID, upnp:scheduledStartTime, upnp:scheduledEndTime
OBJECT.RECORDSCHEDULE.DIRECT.MANUAL	res@dlna:scheduledRecordedContent, dc:date, res@duration, dc:title, upnp:channelID, upnp:scheduledStartTime, upnp:scheduledEndTime
OBJECT.RECORDSCHEDULE.DIRECT.CDSEPG	res@dlna:scheduledRecordedContent, dc:date, res@duration, dc:title, upnp:channelName, upnp:channelNr, upnp:genre, upnp:channelID, upnp:scheduledStartTime, upnp:scheduledEndTime
OBJECT.RECORDSCHEDULE.QUERY.CONTENTNAME	res@dlna:scheduledRecordedContent, dc:date, res@duration, dc:title
OBJECT.RECORDSCHEDULE.QUERY.CONTENTID	res@dlna:scheduledRecordedContent, dc:date, res@duration, dc:title

Table 36 — Recommended Recorded CDS Properties based on srs:class Value

7.4.4.3 MM/SR UPnP ScheduledRecording Service

[GUIDELINE] A UPnP AV MediaServer shall implement the mandatory actions and state variables of a ScheduledRecording service

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[49] [60]	UQXT8	N

7.4.4.4 MM/SR CDS Association

7.4.4.4.1

[GENERAL] This set of guidelines defines the association between the UPnP ScheduledRecording service and the UPnP ContentDirectory service.

7.4.4.4.2

[GUIDELINE] A UPnP AV MediaServer shall implement the DLNA Basic Tuner guidelines defined in 7.4.1.4.15 - 7.4.1.4.22 or the DLNA Extended Tuner guidelines defined in 0 to expose the device’s channel lineup.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46] [49]	TYFVG	N

NOTE: <Delete entire paragraph if there is no note.>

7.4.4.5 MM/SR SRS:GetSortCapabilities Action

7.4.4.5.1

[GUIDELINE] A UPnP AV MediaServer shall respond to the SRS:GetSortCapabilities action with the SortCaps output argument containing the entire list of values that are allowed as defined in [60] in the SortCriteria input argument for any UPnP ScheduledRecording action request.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	DCI2Z	N

NOTE: The UPnP ScheduledRecording service only supports a single mechanism for UPnP AV MediaServers to advertise their support for sorting across the actions defined in the service. Consequently, the UPnP AV MediaServer needs to provide a consistent level of implementation across all actions which accept a SortCriteria input argument so that a ScheduledRecording Controller can rely on the output values of the SRS:GetSortCapabilities across all actions.

7.4.4.5.2

[GUIDELINE] If a UPnP AV MediaServer does not implement sorting in the UPnP ScheduledRecording service, it shall return an empty string in the SortCaps and “0” (zero) in the SortLevelCap output arguments of the SRS:GetSortCapabilities action.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	EAOOX	N

7.4.4.5.3

[GUIDELINE] If a UPnP AV MediaServer includes the value of “srs-conflict-resolution” in the <dlna:X_DLNACAP>, then it shall include the property srs:priority@orderedValue in the SortCaps output argument in response to a SRS:GetSortCapabilities request.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	JJOSD	N

7.4.4.6 MM/SR SRS:BrowseRecordSchedules Action

7.4.4.6.1

[GUIDELINE] If a UPnP AV MediaServer receives one or more property names in the Filter input argument which are not implemented by the UPnP AV MediaServer for the SRS:BrowseRecordSchedules action, then the request shall be processed as if these properties were not included in the Filter input argument. If this results in an empty string, the UPnP AV MediaServer shall only return the required properties as defined in [60] in the result.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	M8LZZ	N

NOTE: This guideline clarifies UPnP AV MediaServer behavior when invalid properties are included in the Filter input argument. A ScheduledRecording Controller can obtain the list of properties supported by the UPnP AV MediaServer using the SRS:GetPropertyList action.

7.4.4.6.2

[GUIDELINE] If a UPnP AV MediaServer receives one or more property names in the SortCriteria input argument for the SRS:BrowseRecordSchedules action which are not returned in the SortCaps output argument in the SRS:GetSortCapabilities action, then it shall return an Error Code 709 (Unsupported or invalid sort criteria).

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	GB2EJ	N

NOTE: This guideline clarifies the UPnP AV MediaServer response to an invalid SortCriteria input argument. A ScheduledRecording Controller can obtain the sort capabilities of the UPnP AV MediaServer by making a SRS:GetSortCapabilities request.

7.4.4.6.3

[GUIDELINE] A UPnP AV MediaServer device may reduce the number of recordSchedule objects in a response to a SRS:BrowseRecordSchedules action for the following scenarios only:

- The transmission of a SOAP response with a huge byte length (>204,800 bytes).
- The transmission of a SOAP response that exceeds 30 seconds for transmission time

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[60]	KOO7Z	N

NOTE: This requirement is to align with the requirement for a SOAP response in CDS:Browse action and CDS:Search action defined in 7.4.1.4.10.1.

7.4.4.6.4

[GUIDELINE] The number of recordSchedule object entries in the Result output argument (containing the XML escaped srs XML Document) shall match the value specified in the NumberReturned output argument.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	98W4S	N

NOTE: This guideline needs to be followed, even if a UPnP AV MediaServer reduces the number of recordSchedule objects returned in the SOAP response. This requirement is to align with the requirement for a SOAP response in CDS:Browse action and CDS:Search action defined in 7.4.1.4.10.2.

7.4.4.6.5

[GUIDELINE] If the UPnP AV MediaServer device cannot find more than zero recordSchedule objects (in 27 seconds, as described in 7.3.2.9.2), for a response to SRS:BrowseRecordSchedules request and if UPnP AV MediaServer cannot calculate an accurate value for the TotalMatches output argument, then the UPnP AV MediaServer should return a SOAP error response code of 720 (Cannot process the request).

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[60]	ZJHU2	N

NOTE: This guideline covers the scenario where a UPnP AV MediaServer can neither find any recordSchedule objects that satisfy the query nor calculate the TotalMatches output argument accurately. Although some UPnP AV MediaServer implementations may choose to report the accurate TotalMatches value, at the expense of violating the 27 seconds timeout rule, such behavior is not recommended for the same reason stated in guideline 7.4.1.4.10.6. This guideline is to align with the guideline for a SOAP response in CDS:Browse action and CDS:Search action defined in 7.4.1.4.10.7.

7.4.4.6.6

[GUIDELINE] A UPnP AV MediaServer control point should specify the desired number of recordSchedule objects in the RequestedCount input argument of a SRS:BrowseRecordSchedules request.

[ATTRIBUTES]

S	C	+SR+	n/a	n/a	[60]	BYNOM	N

NOTE: This guideline recommends control points to request a reasonable number of recordSchedule objects in a single query. The number of recordSchedule objects that can be displayed to the user at a single time is a good measure of reasonableness. Generally speaking, control points that specify smaller RequestedCount values will receive the response from the device sooner than if a larger value were specified. Using a RequestedCount of zero is prohibited by [102]. This guideline is to align with the guideline for a SOAP response in CDS:Browse action and CDS:Search action defined in 7.4.1.4.10.8.

7.4.4.6.7

[GUIDELINE] A UPnP AV MediaServer control point shall tolerate a response with less recordSchedule objects than requested in a SRS:BrowseRecordSchedules request.

[ATTRIBUTES]

M	C	+SR+	n/a	n/a	[60]	5ZXZL	N

NOTE: This guideline provides implementation guidance to a UPnP AV MediaServer control point to not assume that its SRS:BrowseRecordSchedules request will return all of the recordSchedule objects it requested. This requirement is to align with the requirement for a SOAP response in CDS:Browse action and CDS:Search action defined in 7.4.1.4.10.10.

7.4.4.6.8

[GUIDELINE] A UPnP AV MediaServer control point should retrieve the remaining items in a reduced response to a SRS:BrowseRecordSchedules request, when the value of TotalMatches is greater than the value of NumberReturned, by issuing additional SRS:BrowseRecordSchedules requests to complete the original SRS:BrowseRecordSchedules request for recordSchedule objects.

[ATTRIBUTES]

S	C	+SR+	n/a	n/a	[60]	I7UFV	N

NOTE: This guideline requirement provides implementation guidance to UPnP AV MediaServer control points when a UPnP AV MediaServer returns more than zero recordSchedule objects in a response to a SRS:BrowseRecordSchedules action request with a reduced response. This requirement is to align with the requirement for a SOAP response in CDS:Browse action and CDS:Search action defined in 7.4.1.4.10.11.

7.4.4.7 MM/SR BrowseRecordTasks Action

7.4.4.7.1

[GUIDELINE] If a UPnP AV MediaServer receives one or more property names in the Filter input argument which are not implemented by the UPnP AV MediaServer for the SRS:BrowseRecordTasks action, then the request shall be processed as if these properties were not included in the Filter input argument. If this results in an empty string, the UPnP AV MediaServer shall only return the required properties as defined in [60] in the result.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	RRNYM	N

NOTE: This guideline clarifies UPnP AV MediaServer behavior when invalid properties are included in the Filter input argument. A ScheduledRecording Controller can obtain the list of properties supported by the UPnP AV MediaServer using the SRS:GetPropertyList action.

7.4.4.7.2

[GUIDELINE] If a UPnP AV MediaServer receives one or more property names in the SortCriteria input argument for the SRS:BrowseRecordTasks action which are not returned in the SortCaps output argument in the SRS:GetSortCapabilities action , then it shall return an Error Code 709 (Unsupported or invalid sort criteria).

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	RUICO	N

NOTE: This guideline clarifies the UPnP AV MediaServer response to an invalid SortCriteria input argument. A ScheduledRecording Controller can obtain the sort capabilities of the UPnP AV MediaServer by making a SRS:GetSortCapabilities request.

7.4.4.7.3

[GUIDELINE] A UPnP AV MediaServer device may reduce the number of recordTask objects in a response to a SRS:BrowseRecordTasks action for the following scenarios only:

- The transmission of a SOAP response with a huge byte length (>204,800 bytes).
- The transmission of a SOAP response that exceeds 30 seconds for transmission time.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[60]	572H8	N

NOTE: This guideline is to align with the requirement for a SOAP response in CDS:Browse action and CDS:Search action defined in 7.4.1.4.10.1.

7.4.4.7.4

[GUIDELINE] The number of recordTask object entries in the Result output argument (containing the XML escaped srs XML Document) shall match the value specified in the NumberReturned output argument.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	6YLEZ	N

NOTE: This guideline needs to be followed, even if a UPnP AV MediaServer reduces the number of recordTask objects returned in the SOAP response. This guideline is to align with the guideline for a SOAP response in CDS:Browse action and CDS:Search action defined in 7.4.1.4.10.2.

7.4.4.7.5

[GUIDELINE] If the UPnP AV MediaServer device cannot find more than zero recordTask objects (in 27 seconds, as described in 7.3.2.9.2), for a response to SRS:BrowseRecordTasks request and if UPnP AV MediaServer cannot calculate an accurate value for the TotalMatches output argument, then the UPnP AV MediaServer should return a SOAP error response code of 720 (Cannot process the request).

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[60]	683H9	N

NOTE: This guideline covers the scenario where a UPnP AV MediaServer can neither find any recordTask objects that satisfy the query nor calculate the TotalMatches output argument accurately. Although some UPnP AV MediaServer implementations may choose to report the accurate TotalMatches value, at the expense of violating the 27 seconds timeout rule, such behavior is not recommended for the same reason stated in guideline 7.4.1.4.10.6. This guideline is to align with the requirement for a SOAP response in CDS:Browse action and CDS:Search action defined in 7.4.1.4.10.7.

7.4.4.7.6

[GUIDELINE] A UPnP AV MediaServer control point should specify the desired number of recordTask objects in the RequestedCount input argument of a SRS:BrowseRecordTasks request.

[ATTRIBUTES]

S	C	+SR+	n/a	n/a	[60]	9QCYT	N

NOTE: This guideline recommends control points to request a reasonable number of recordTask objects in a single query. The number of recordTask objects that can be displayed to the user at a single time is a good measure of reasonableness. Generally speaking, control points that specify smaller RequestedCount values will receive the response from the device sooner than if a larger value were specified. Using a RequestedCount of zero is prohibited by [102]. This requirement is to align with the guideline for a SOAP response in CDS:Browse action and CDS:Search action defined in 7.4.1.4.10.8.

7.4.4.7.7

[GUIDELINE] A UPnP AV MediaServer control point shall tolerate a response with less recordTask objects than requested in a SRS:BrowseRecordTasks request.

[ATTRIBUTES]

M	C	+SR+	n/a	n/a	[60]	OICAS	N

NOTE: This guideline requirementprovides implementation guidance to a UPnP AV MediaServer control point to not assume that its SRS:BrowseRecordTasks request will return all of the recordTask objects it requested. This requirement is to align with the requirement for a SOAP response in CDS:Browse action and CDS:Search action defined in 7.4.1.4.10.10.

7.4.4.7.8

[GUIDELINE] A UPnP AV MediaServer control point should retrieve the remaining items in a reduced response to a SRS:BrowseRecordTasks request, when the value of TotalMatches is greater than the value of NumberReturned, by issuing additional SRS:BrowseRecordTasks requests to complete the original SRS:BrowseRecordTasks request for recordTask objects.

[ATTRIBUTES]

S	C	+SR+	n/a	n/a	[60]	9AXKX	N

NOTE: This guideline requirement provides implementation guidance to UPnP AV MediaServer control points when a UPnP AV MediaServer returns more than zero recordSchedule objects in a response to a CDS:BrowseRecordTasks action request with a reduced response. This requirement is to align with the requirement for a SOAP response in CDS:Browse action and CDS:Search action defined in 7.4.1.4.10.11.

7.4.4.8

7.4.4.8.1 MM/SR Representation of Allowed Values Description

[GENERAL] These guideline allow a UPnP AV MediaServer to reduce the allowed values description returned in the PropertyInfo output argument from SRS:GetAllowedValues action. The allowed values description is described as an XML document with the format (syntax) specified in [61].

7.4.4.8.2

[GUIDELINE] The allowed values returned by the UPnP AV MediaServer in response to a SRS:GetAllowedValues request may contain <allowedValueDescriptor> elements which omit some <dependentField> sub elements.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[60] [61]	BXUSP	N

NOTE: The dependencies between certain properties are defined in the UPnP SRS Specification [60]. Therefore the <dependentField> sub element is not always necessary in the <allowedValueDescriptor> elements for these properties. Vendors are strongly encouraged to omit unnecessary <dependentField> sub elements to minimize the length of the output argument to SRS:GetAllowedValues.

The following examples show two cases in which some <dependentField> sub elements could be omitted:

- This example illustrates the allowed values description for the srs:scheduledCDSObjectID property.

```
<field>
   <name>srs:scheduledCDSObjectID</name>
```

```
      <dataType maxSize="1024">xsd:string</dataType>
      <allowedValueDescriptor>
         <dependentField>
            <name>srs:class</name>
            <valueList>
               <value>OBJECT.RECORDSCHEDULE.DIRECT.CDSEPG</value>
               <value>OBJECT.RECORDSCHEDULE.DIRECT.CDSNONEPG</value>
            </valueList>
         </dependentField>
         <minCount>1</minCount>
         <allowAny/>
      </allowedValueDescriptor>
   </field>
```

The SRS specification [60] defines the srs:scheduledCDSObjectID property to be used only with the cdsNonEPG and cdsEPG record classes. Therefore the above XML fragment can be reduced to the following:

```
   <field>
      <name>srs:scheduledCDSObjectID</name>
      <dataType maxSize="1024">xsd:string</dataType>
      <allowedValueDescriptor>
         <minCount>1</minCount>
         <allowAny/>
      </allowedValueDescriptor>
   </field>
```

- This example illustrates the allowed values description for the srs:desiredPriority@type property

```
   <field>
      <name>srs:desiredPriority@type</name>
      <dataType maxSize="16">xsd:string</dataType>
      <allowedValueDescriptor>
         <dependentField>
            <name>srs:desiredPriority</name>
            <anyValue/>
         </dependentField>
         <minCount>1</minCount>
         <allowedValueList>
            <allowedValue>PREDEF</allowedValue>
            <allowedValue>OBECTID</allowedValue>
         </allowedValueList>
      </allowedValueDescriptor>
   </field>
```

An XML attribute can only exist in the context of its parent element. Therefore the above XML fragment can be reduced to the following:

```
   <field>
      <name>srs:desiredPriority@type</name>
      <dataType maxSize="16">xsd:string</dataType>
      <allowedValueDescriptor>
         <minCount>1</minCount>
         <allowedValueList>
            <allowedValue>PREDEF</allowedValue>
            <allowedValue>OBECTID</allowedValue>
         </allowedValueList>
      </allowedValueDescriptor>
   </field>
```

7.4.4.8.3

[Guideline] A UPnP AV MediaServer control point shall be able to parse and interpret <allowedValueDescriptor> elements that omit the <dependentField> sub element.

[Attributes]

M	C	+SR+	n/a	n/a	[60] [61]	55STS	N

NOTE: Guideline requirement 7.4.4.8.2 allows UPnP AV MediaServers to omit the <dependentField> sub elements.

7.4.4.9 MM/SR cdsNonEPG Record Class

7.4.4.9.1

[General] This defines the guidelines for a UPnP AV MediaServer when implementing the mandatory cdsNonEPG record class for the ScheduledRecording service.

7.4.4.9.2

[Guideline] A UPnP AV MediaServer shall respond at a minimum with the value of "OBJECT.RECORDSCHEDULE.DIRECT.CDSNONEPG" to the SRS:GetAllowedValues action for the srs:class property when the DataTypeID input argument is the value A_ARG_TYPE_RecordScheduleParts or A_ARG_TYPE_RecordSchedule

[Attributes]

M	R	DMS	M-DMS	n/a	[60]	9JGNA	N

NOTE: This reiterates the mandatory record class that's implemented by a UPnP AV MediaServer with a ScheduledRecording service.

7.4.4.9.3

[Guideline] A UPnP AV MediaServer shall respond to the SRS:GetAllowedValues action for the srs:scheduledCDSObjectID property when the DataTypeID input argument is the value A_ARG_TYPE_RecordScheduleParts or A_ARG_TYPE_RecordSchedule with either the following:

- A list of all of the CDS @id values, that are available to setup a RecordSchedule
- A value of "<allowAny></allowAny>

[Attributes]

M	C	DMS	M-DMS	n/a	[60]	I9VW2	N

NOTE: The UPnP AV MediaServer uses the value "<allowAny></allowAny>" to indicate that it allows any CDS object with the upnp:recordable property set to 1.

7.4.4.9.4

[Guideline] If a UPnP AV MediaServer responds to the SRS:GetAllowedValues action for the srs:scheduledCDSObjectID property with a list of the CDS @id values, then the corresponding CDS items shall exist and shall have the upnp:recordable property set to "1"

[Attributes]

M	C	DMS	M-DMS	n/a	[60]	OEABB	N

7.4.4.9.5

[Guideline] A UPnP AV MediaServer should set the value of the dc:title property in the recorded content CDS item with the same value as the dc:title property contained in the CDS item specified by the srs:scheduledCDSObjectID property, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledCDSObjectID property is from the recordSchedule object that spawned the recordTask.

[Attributes]

S	A	DMS	M-DMS	n/a	[46]	YTG5V	N

NOTE: This guideline requirement is semantically equivalent to guideline 7.4.4.11.6 for the cdsEPG record class.

7.4.4.9.6

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:channelName property in the recorded content CDS item with the same value as the upnp:channelName property contained in the CDS item specified by the srs:scheduledCDSObjectID property, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask
- The srs:scheduledCDSObjectID property is from the recordSchedule object that spawned the recordTask.
- The CDS item specified by the srs:scheduedCDSObjectID property exposes the upnp:channelName property.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	SUQGU	N

NOTE: This guideline is semantically equivalent to guideline7.4.4.11.7 for the cdsEPG record class.

7.4.4.9.7

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:channelNr property in the recorded content CDS item with the same value as the upnp:channelNr property contained in the CDS item specified by the srs:scheduledCDSObjectID property, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledCDSObjectID property is from the recordSchedule object that spawned the recordTask.
- The CDS item specified by the srs:scheduedCDSObjectID property exposes the upnp:channelNr property.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	72MV6	N

NOTE: This guideline is semantically equivalent to guideline 7.4.4.11.8 for the cdsEPG record class.

7.4.4.9.8

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:genre property in the recorded content CDS item with the same value as the upnp:genre property contained in the CDS item specified by the srs:scheduledCDSObjectID property, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledCDSObjectID property is from the recordSchedule object that spawned the recordTask.
- The CDS item specified by the srs:scheduedCDSObjectID property exposes the upnp:genre property.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	GMGFV	N

NOTE: This guideline is semantically equivalent to guideline 7.4.4.11.9 for the cdsEPG record class.

7.4.4.9.9

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:channelID and upnp:channelID@type properties in the recorded content CDS item with the same values as the upnp:channelID and upnp:channelID@type properties contained in the CDS item specified by the srs:scheduledCDSObjectID property, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledCDSObjectID property is from the recordSchedule object that spawned the recordTask.
- The CDS item specified by the srs:scheduedCDSObjectID property exposes the upnp:channelID and upnp:channelID@type properties.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	GPARX	N

NOTE: This guideline requirement is semantically equivalent to guideline requirements 7.4.4.10.5 and 7.4.4.11.10 for the Manual and cdsEPG record classes respectively.

7.4.4.9.10

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:scheduledStartTime property in the recorded content CDS item with the same value as the srs:scheduledStartDateTime property contained in the recordSchedule object, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledStartDateTime property is from the recordSchedule object that spawned the recordTask.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	RZSWF	N

NOTE: This value is not necessarily the actual start time of the recording, that value would be contained in the dc:date property.

This guideline is semantically equivalent to guidelines 7.4.4.10.6 and 7.4.4.11.11 for the Manual and cdsEPG record classes respectively.

7.4.4.9.11

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:scheduledEndTime property in the recorded content CDS item with the same value as the sum of the srs:scheduledStartDateTime and srs:scheduledDuration properties contained in the recordSchedule object, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledStartDateTime and srs:scheduledDuration properties are from the recordSchedule object that spawned the recordTask.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	9N6I9	N

NOTE: This value is not necessarily the actual end time of the recording, that value can be calculated by summing the dc:date and res@duration properties.

This guideline is semantically equivalent to guidelines 7.4.4.10.7 and 7.4.4.11.12 for the Manual and cdsEPG record classes respectively.

7.4.4.10 MM/SR Manual Record Class

7.4.4.10.1

[General] This defines the requirements for a UPnP AV MediaServer when implementing the optional manual record class for the ScheduledRecording service

7.4.4.10.2

[Guideline] If a UPnP AV MediaServer implements the manual record class, then it shall include in the response the value of "OBJECT.RECORDSCHEDULE.DIRECT.MANUAL" to the SRS:GetAllowedValues action for the srs:class property when the DataTypeID input argument is the value A_ARG_TYPE_RecordScheduleParts or A_ARG_TYPE_RecordSchedule.

[Attributes]

M	R	DMS	M-DMS	n/a	[60]	8UMKG	N

7.4.4.10.3

[Guideline] If a UPnP AV MediaServer indicates support for the manual record class as defined in guideline 7.4.4.10.2, then a UPnP AV MediaServer shall respond to the SRS:GetAllowedValues action for the srs:scheduledChannelID property when the DataTypeID input argument is the value A_ARG_TYPE_RecordScheduleParts or A_ARG_TYPE_RecordSchedule with either the following:

- A list of all of the CDS channelID values, that are available to setup a RecordSchedule
- A value of "<allowAny></allowAny>"

[Attributes]

M	C	DMS	M-DMS	n/a	[60]	C4QMW	N

NOTE: The UPnP AV MediaServer uses the value "<allowAny></allowAny>" to indicate that it allows any upnp:channelID value obtained from a Tuner channel item with the upnp:recordable property set to 1.

7.4.4.10.4

[Guideline] If a UPnP AV MediaServer indicates support for the manual record class as defined in guideline requirement 7.4.4.10.2 and responds to the SRS:GetAllowedValues action for the srs:scheduledChannelID property with a list of the channelID values, then the corresponding CDS items containing those upnp:channelID properties shall have the upnp:recordable property set to "1".

[Attributes]

M	C	DMS	M-DMS	n/a	[60]	D2XAU	N

7.4.4.10.5

[Guideline] A UPnP AV MediaServer should expose the upnp:channelID and upnp:channelID@type properties in the recorded content CDS item with the same values as the srs:scheduledChannelID and srs:scheduledChannelID@type properties contained in the recordSchedule object, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.

- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledChannelID and srs:scheduledChannelID@type properties are from the recordSchedule object that spawned the recordTask.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/s	[46]	W5Q29	N

NOTE: This guideline is semantically equivalent to guidelines 7.4.4.9.9 and 7.4.4.11.10 for the cdsNonEPG and cdsEPG record classes respectively.

7.4.4.10.6

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:scheduledStartTime property in the recorded content CDS item with the same value as the srs:scheduledStartDateTime property contained in the recordSchedule object, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledStartDateTime property is from the recordSchedule object that spawned the recordTask.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/s	[46]	YSN9Y	N

NOTE: The value is not necessarily the actual start time of the recording, that value would be contained in the dc:date property.

This guideline requirement is semantically equivalent to guideline requirements 7.4.4.9.10 and 7.4.4.11.11 for the cdsNonEPG and cdsEPG record classes respectively.

7.4.4.10.7

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:scheduledEndTime property in the recorded content CDS item with the same value as the sum of the srs:scheduledStartDateTime and srs:scheduledDuration properties contained in the recordSchedule object, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledStartDateTime and srs:scheduledDuration properties are from the recordSchedule object that spawned the recordTask.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/s	[46]	RYLAY	N

NOTE: This value is not necessarily the actual end time of the recording, that value can be calculated by summing the dc:date and res@duration properties.

This guideline requirement is semantically equivalent to guideline requirements 7.4.4.9.11 and 7.4.4.11.12 for the cdsNonEPG and cdsEPG record classes respectively.

7.4.4.11 MM/SR cdsEPG Record Class

7.4.4.11.1

[GENERAL] This defines the requirements for a UPnP AV MediaServer when implementing the optional cdsEPG record class for the ScheduledRecording service.

7.4.4.11.2

[GUIDELINE] If a UPnP AV MediaServer implements the cdsEPG record class as defined in guidelines 7.4.4.11.3 - 7.4.4.11.5 respectively, then it shall implement the EPG Server Device Option, as defined in 7.4.6.2.3, by including a <Feature> element with the Feature@name attribute equal to "EPG" in the FeatureList output argument in response to the CDS:GetFeatureList action.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [60]	WE94G	N

7.4.4.11.3

[GUIDELINE] If a UPnP AV MediaServer implements the cdsEPG record class, then it shall respond with the value of "OBJECT.RECORDSCHEDULE.DIRECT.CDSEPG" to the SRS:GetAllowedValues action for the srs:class property when the DataTypeID input argument is the value A_ARG_TYPE_RecordScheduleParts or A_ARG_TYPE_RecordSchedule.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[60]	74ODB	N

7.4.4.11.4

[GUIDELINE] If a UPnP AV MediaServer indicates support for the cdsEPG record class as defined in guideline 7.4.4.11.3, then a UPnP AV MediaServer shall respond to the SRS:GetAllowedValues action for the srs:scheduledCDSObjectID property when the DataTypeID input argument is the value A_ARG_TYPE_RecordScheduleParts or A_ARG_TYPE_RecordSchedule with either the following:

- A list of all of the CDS @id values, that are available to setup a recordSchedule
- A value of "<allowAny></allowAny>"

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	D92QI	N

NOTE: The UPnP AV MediaServer uses the value "<allowAny></allowAny>" to indicate that it allows any CDS object with the upnp:class property set to object.item.epgItem or its derived classes and the upnp:recordable property set to 1.

7.4.4.11.5

[GUIDELINE] If a UPnP AV MediaServer indicates support for the cdsEPG record class as defined in guideline 7.4.4.11.3 and responds to the SRS:GetAllowedValues action for the srs:scheduledCDSObjectID property with a list of the CDS @id values, then the corresponding CDS items shall exist, shall have the upnp:class property set to object.item.epgItem or its derived classes, and shall have the upnp:recordable property set to "1".

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	NO8L4	N

NOTE: <Delete entire paragraph if there is no note.>

7.4.4.11.6

[GUIDELINE] A UPnP AV MediaServer should set the value of the dc:title property in the recorded content CDS item with the same value as the dc:title property contained in the CDS item specified by the srs:scheduledCDSObjectID property, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledCDSObjectID property is from the recordSchedule object that spawned the recordTask.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	HPIW2	N

NOTE: This guideline is semantically equivalent to guideline 7.4.4.9.5 for the cdsNonEPG record class.

7.4.4.11.7

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:channelName property in the recorded content CDS item with the same value as the upnp:channelName property contained in the CDS item specified by the srs:scheduledCDSObjectID property, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledCDSObjectID property is from the recordSchedule object that spawned the recordTask.
- The CDS item specified by the srs:schedued CDSObjectID property exposes the upnp:channelName property.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	G3MOE	N

NOTE: This guideline is semantically equivalent to guideline 7.4.4.9.6 for the cdsNonEPG record class.

7.4.4.11.8

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:channelNr property in the recorded content CDS item with the same value as the upnp:channelNr property contained in the CDS item specified by the srs:scheduledCDSObjectID property, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledCDSObjectID property is from the recordSchedule object that spawned the recordTask.
- The CDS item specified by the srs:scheduedCDSObjectID property exposes the upnp:channelNr property.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	5H8V6	N

NOTE: This guideline is semantically equivalent to guideline 7.4.4.9.7 for the cdsNonEPG record class.

7.4.4.11.9

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:genre property in the recorded content CDS item with the same value as the upnp:genre property contained in the CDS item specified by the srs:scheduledCDSObjectID property, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.

- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledCDSObjectID property is from the recordSchedule object that spawned the recordTask.
- The CDS item specified by the srs:schedued CDSObjectID property exposes the upnp:genre property.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	5K399	N

NOTE: This guideline is semantically equivalent to guideline 7.4.4.9.8 for the cdsNonEPG record class.

7.4.4.11.10

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:channelID and upnp:channelID@type properties in the recorded content CDS item with the same values as the upnp:channelID and upnp:channelID@type properties contained in the CDS item specified by the srs:scheduledCDSObjectID property, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledCDSObjectID property is from the recordSchedule object that spawned the recordTask.
- The CDS item specified by the srs:schedued CDSObjectID property exposes the upnp:channelID and upnp:channelID@type properties.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	32SPN	N

NOTE: This guideline is semantically equivalent to guidelines 7.4.4.9.9 and 7.4.4.10.5 for the cdsNonEPG and Manual record classes respectively

7.4.4.11.11

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:scheduledStartTime property in the recorded content CDS item with the same value as the upnp:scheduledStartTime property contained in the CDS item specified by the srs:scheduledCDSObjectID property, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledCDSObjectID property is from the recordSchedule object that spawned the recordTask.
- The CDS item specified by the srs:schedued CDSObjectID property exposes the upnp:scheduledStartTime property.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	4UDNI	N

NOTE: This value is not necessarily the actual start time of the recording, that value would be contained in the dc:date property.

This guideline is semantically equivalent to guidelines 7.4.4.9.10 and 7.4.4.10.6 for the cdsNonEPG and Manual record classes respectively.

7.4.4.11.12

[GUIDELINE] A UPnP AV MediaServer should expose the upnp:scheduledEndTime property in the recorded content CDS item with the same value as the upnp:scheduledEndTime property contained in the CDS item specified by the srs:scheduledCDSObjectID property, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.
- The srs:scheduledCDSObjectID property is from the recordSchedule object that spawned the recordTask.
- The CDS item specified by the srs:scheduedCDSObjectID property exposes the upnp:scheduledEndTime property.

[Attributes]

S	A	DMS	M-DMS	n/a	[46]	VF9KL	N

NOTE: This value is not necessarily the actual end time of the recording, that value can be calculated from by summing the dc:date and res@duration properties.

This guideline is semantically equivalent to guidelines 7.4.4.9.11 and 7.4.4.10.7 for the cdsNonEPG and Manual record classes respectively.

7.4.4.12 MM/SR Query Content Name Record Class

7.4.4.12.1

[General] This defines the guidelines for a UPnP AV MediaServer when implementing the optional query contentName record class for the ScheduledRecording Service

7.4.4.12.2

[Guideline] If a UPnP AV MediaServer implements the query content name record class, then it shall respond with the value of “OBJECT.RECORDSCHEDULE.QUERY.CONTENTNAME” to the SRS:GetAllowedValues action for the srs:class property when the input parameter DataTypeID is the value A_ARG_TYPE_RecordScheduleParts or A_ARG_TYPE_RecordSchedule.

[Attributes]

M	R	DMS	M-DMS	n/a	[60]	YXJ47	N

7.4.4.12.3

[Guideline] If a UPnP AV MediaServer indicates support for the query content name record class as defined in guideline requirement 7.4.4.12.2, then a UPnP AV MediaServer shall respond to the SRS:GetAllowedValues action for the srs:matchingName property when the input parameter DataTypeID contains the value A_ARG_TYPE_RecordScheduleParts or A_ARG_TYPE_RecordSchedule with the following:

- A value of “<allowAny></allowAny>”

[Attributes]

M	C	DMS	M-DMS	n/a	[60]	ZVPQ5	N

7.4.4.12.4

[Guideline] If a UPnP AV MediaServer indicates support for the query content name record class as defined in guideline requirement 7.4.4.12.2, then a UPnP AV MediaServer shall respond to the SRS:GetAllowedValues action for the srs:matchingName@type property when the input parameter DataTypeID contains the value A_ARG_TYPE_RecordScheduleParts or A_ARG_TYPE_RecordSchedule with at least one of the following:

- A value of “PROGRAM”
- A value of “SERIES”

[Attributes]

M	C	DMS	M-DMS	n/a	[60]	RCX7F	N

7.4.4.12.5

[Guideline] A UPnP AV MediaServer should set the value of the dc:title property in the recorded content CDS item with the title of the program recorded, when all of the following conditions are met:

[Attributes]

S	A	DMS	M-DMS	n/a	[46]	TZUF6	N

NOTE: For the OBJECT.RECORDSCHEDULE.QUERY.CONTENTNAME record class, the existence of the name of the program depends on the external databases (like Service Information) and it might not be available in a CDS item.

This guideline requirement is semantically equivalent to guideline requirement 7.4.4.13.5 for the Query Content ID record class.

7.4.4.13 MM/SR Query Content ID Record Class

7.4.4.13.1

[General] This defines the requirements for a UPnP AV MediaServer when implementing the optional query contentID record class for the ScheduledRecording Service.

7.4.4.13.2

[Guideline] If a UPnP AV MediaServer implements the query content ID record class, then it shall respond with the value of “OBJECT.RECORDSCHEDULE.QUERY.CONTENTID” to the SRS:GetAllowedValues action for the srs:class property when the input parameter DataTypeID is the value A_ARG_TYPE_RecordScheduleParts or A_ARG_TYPE_RecordSchedule.

[Attributes]

M	R	DMS	M-DMS	n/a	[60]	GTDQA	N

7.4.4.13.3

[Guideline] If a UPnP AV MediaServer indicates support for the query content ID record class as defined in guideline requirement 7.4.4.13.2, then a UPnP AV MediaServer shall respond to the SRS:GetAllowedValues action for the srs:matchingID property when the input parameter DataTypeID contains the value A_ARG_TYPE_RecordScheduleParts or A_ARG_TYPE_RecordSchedule with the following:

- A value of “<allowAny></allowAny>”

[Attributes]

M	C	DMS	M-DMS	n/a	[60]	L8ZJP	N

7.4.4.13.4

[Guideline] If a UPnP AV MediaServer indicates support for the query content name record class as defined in guideline requirement 7.4.4.13.2, then a UPnP AV MediaServer shall respond to the SRS:GetAllowedValues action for the srs:matchingID@type property when the input parameter DataTypeID contains the value A_ARG_TYPE_RecordScheduleParts or A_ARG_TYPE_RecordSchedule with at least one of the following:

- A value of “SI_PROGRAMID”
- A value of “SI_SERIESID”

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	G5O6J	N

7.4.4.13.5

[GUIDELINE] A UPnP AV MediaServer should set the value of the dc:title property in the recorded content CDS item with the title of the program recorded, when all of the following conditions are met:

- The UPnP AV MediaServer exposes recorded content in a ContentDirectory service.
- The recorded content is a result of the execution of a recordTask.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	23T7R	N

NOTE: For the OBJECT.RECORDSCHEDULE.QUERY.CONTENTID record class, the existence of the name of the program depends on the external databases (like Service Information) and it might not be available in a CDS item.

This guideline is semantically equivalent to guideline 7.4.4.12.5 for the Query Content Name record class

7.4.4.14 MM/SR Query Record Class and EPG

7.4.4.14.1

[GENERAL] This defines the guidelines for a UPnP AV MediaServer when implementing the optional query contentName or contentID record classes and its interaction with the EPG Server Device Option.

7.4.4.14.2

[GUIDELINE] If a UPnP AV MediaServer implements the query contentName or query contentID record class as defined in guideline requirements 7.4.4.12 and 7.4.4.13 respectively, then it should implement the EPG Server Device Option, as defined 7.4.6.2.3, by including a <Feature> element with the Feature@name attribute equal to “EPG” in the FeatureList output argument in response to the CDS:GetFeatureList action

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46] [60]	CIZ2D	N

NOTE: It is also possible for a UPnP AV MediaServer to have internally maintained EPG information that is not exposed by the CDS. In such a case this <Feature> element is not used.

7.4.4.14.3

[GUIDELINE] If a UPnP AV MediaServer implements the EPG Server Device Option as defined in 7.4.4.14.2 and returns the value of “PROGRAM” in the response to the SRS:GetAllowedValues action for the srs:matchingName@type property as defined in 7.4.4.12.4, then it should expose the upnp:programTitle property for one or more current or future EPG Program Items.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[60]	6KBDC	N

NOTE: The object.recordSchedule.query.ContentName recording class requires a Series or Program Title as the match string input. The query is done by matching the upnp:programTitle property, and that there is no way a matching EPG program item would be found if the property is not exposed. Hence it is strongly recommended that upnp:programTitle property is exposed.

7.4.4.14.4

[GUIDELINE] If a UPnP AV MediaServer implements the EPG Server Device Option as defined in 7.4.4.14.2 and returns the value of “SERIES” in the response to the SRS:GetAllowedValues action for the srs:matchingName@type property as defined in 7.4.4.12.4, then it should expose the upnp:seriesTitle property for one or more current or future EPG Program Items.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[60]	IQ7RP	N

NOTE: The object.recordSchedule.query.ContentName recording class requires a Series or 7Program Title as the match string input. The query is done by matching the upnp:seriesTitle property, and that there is no way a matching EPG program item would be found if the property is not exposed. Hence it is strongly recommended that upnp:seriesTitle property is exposed

7.4.4.14.5

[GUIDELINE] If a UPnP AV MediaServer implements the EPG Server Device Option as defined in 7.4.4.14.2 and returns the value of “SI_PROGRAMID” in the response to the SRS:GetAllowedValues action for the srs:matchingID@type property as defined in 7.4.4.13.4, then it should expose the upnp:programID property for one or more current or future EPG Program Items.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[60]	SBZCD	N

NOTE: The object.recordSchedule.query.ContentID recording class requires a Series or Program ID as the match string input. The query is done by matching the upnp:programID property, and that there is no way a matching EPG program item would be found if the property is not exposed. Hence it is strongly recommended that upnp:programID property is exposed

7.4.4.14.6

[GUIDELINE] If a UPnP AV MediaServer implements the EPG Server Device Option as defined in 7.4.4.14.2 and returns the value of “SI_SERIESID” in the response to the SRS:GetAllowValues action for the srs:matchingID@type property as defined 7.4.4.13.4, then it should expose the upnp:seriesID property for one or more current or future EPG Program Items.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[60]	SETOI	N

NOTE: The object.recordSchedule.query.ContentID recording class requires a Series or Program ID as the match string input. The query is done by matching the upnp:SeriesID property, and that there is no way a matching EPG program item would be found if the property is not exposed. Hence it is strongly recommended that upnp:seriesID property is exposed.

7.4.4.15 MM/SR Conflict Resolution

7.4.4.15.1

[GENERAL] This describes the general guidelines for supporting Conflict Resolution.

7.4.4.15.2

[GUIDELINE] If a UPnP AV MediaServer allows the ScheduledRecording control points to resolve conflicts, then it shall use the <dlna:X_DLNACAP> element (as a child of the <device> element that represents the UPnP AV MediaServer) in the device description document and include the Capability ID “srs-conflict-resolution” in the element's comma-separated value list.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	48KL9	N

NOTE: DMS devices use the <dlna:X_DLNACAP> element to indicate support for SRS conflict resolution operation. The element is a comma separated value list that indicates whether the DMS can resolve schedule conflicts, receive uploads of images, audio-only, or audio/video content etc. See guideline 7.3.2.35.1 for the formal syntax of the <dlna:X_DLNACAP> element. A sample description is given below:

<dlna:X_DLNACAP xmlns:dlna="urn:schemas-dlna-org:device-1-0">image-upload,audio-upload,srs-conflict-resolution,srs-cr-partial-recording</dlna:X_DLNACAP>

7.4.4.15.3

[GUIDELINE] A UPnP AV MediaServer shall implement the SRS:GetRecordScheduleConflicts and SRS:GetRecordTaskConflicts actions.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	625J2	N

NOTE: Since there is guideline 7.4.4.16.7 that mandates the creation of conflicting recordSchedule(s), the UPnP AV MediaServer needs to implement these two actions.

7.4.4.15.4

[GUIDELINE] If a UPnP AV Media Server contains the value of "srs-conflict-resolution" in the <dlna:X_DLNACAP> then it shall implement the following actions:

- SRS:EnableRecordSchedule,
- SRS:DisableRecordSchedule,
- SRS:DeleteRecordTask,
- SRS:EnableRecordTask,
- SRS:DisableRecordTask,
- SRS:ResetRecordTask

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	J3E9B	N

7.4.4.15.5

[GUIDELINE] A UPnP AV MediaServer should enable partial recording of a conflict loser recordTask for the duration that is not conflicting with any other recordTask or a program winner for that duration.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[60]	8RV4U	N

NOTE: A UPnP AV MediaServer need to record the portions of the programs described in the shaded regions.

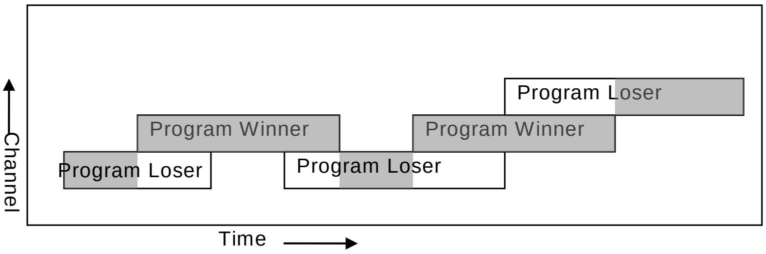


Figure 29 — Recording Conflict Behavior

7.4.4.15.6

[GUIDELINE] If a UPnP AV MediaServer allows partial recordings as described in 7.4.4.15.5, then it shall use the <dlna:X_DLNACAP> element (as a child of the <device> element that represents the UPnP AV MediaServer) in the device description document and include the Capability ID “srs-cr-partial-recording” in the element's comma-separated value list

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	3DN2V	N

NOTE: Partial recordings can also result from other reasons, such as for example, SRS:DisableRecordTask or SRS:DeleteRecordTask actions on a recordTask that is in “ACTIVE” state. Those are not covered by this attribute.

DMS devices use the <dlna:X_DLNACAP> element to indicate support for SRS partial recording operation. The element is a comma separated value list that indicates whether the DMS can create partial recordings, receive uploads of images, audio-only, or audio/video content etc. See guideline 7.3.2.35.1 for the formal syntax of the <dlna:X_DLNACAP> element. A sample description is given below:

```
<dlna:X_DLNACAP xmlns:dlna="urn:schemas-dlna-org:device-1-0">image-upload,audio-
upload,srs-conflict-resolution,srs-cr-partial-recording</dlna:X_DLNACAP>
```

7.4.4.15.7

[GUIDELINE] A UPnP AV MediaServer that includes a value of “srs-conflict-resolution” in the <dlna:X_DLNACAP> shall implement and expose the property srs:priority@orderedValue.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	SSUI7	N

7.4.4.15.8

[GUIDELINE] When a UPnP AV MediaServer creates or modifies a recordTask and if that results in conflict(s) with one or more recordTask(s), then the UPnP AV MediaServer should add the value 402 (Conflicting Program Winner) to the winning recordTask’s srs:taskState@infoList property.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[60]	VHRPT	N

7.4.4.15.9

[GUIDELINE] If a UPnP AV Media Server creates or modifies a recordTask and if that results in conflict(s) with one or more recordTask(s), then the UPnP AV MediaServer shall add the value 401 (Conflicting Program Loser) to the srs:taskState@pendingErrors property of each of the losing recordTask(s). In addition, for each of the recordTask(s) that will be partially recorded, the UPnP AV MediaServer that includes “srs-cr-partial-recording” in <dlna:X_DLNACAP> shall add the value 450 (DLNA Conflicting Partial Program Loser) to the srs:taskState@infoList property.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	4HIEY	N

7.4.4.15.10

[GUIDELINE] When a UPnP AV MediaServer creates or modifies a recordTask and if that causes one or more active recordTask(s) to be stopped or suspended due to the conflict, then for each of those recordTask(s) the UPnP AV MediaServer shall add the value 401 (Conflicting Program Loser) to the srs:taskState@currentErrors property. In addition, for each of the recordTask(s) that will be partially recorded, the UPnP AV MediaServer that includes “srs-cr-partial-recording” in <dlna:X_DLNACAP> shall add the value 450 (DLNA Conflicting Partial Program Loser) to the srs:taskState@infoList property.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[60]	MOWUH	N

7.4.4.15.11

[GUIDELINE] If a UPnP AV MediaServer creates or modifies a recordSchedule, it should create recordTask object(s) from the point it has all the necessary information in a reasonable time.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[60]	HLLHB	N

7.4.4.16 MM/SR SRS:CreateRecordSchedule Action

7.4.4.16.1

[GUIDELINE] If a UPnP AV MediaServer control point sends a SRS:CreateRecordSchedule request where the Elements input argument has the srs:class property with a value of “OBJECT.RECORDSCHEDULE.DIRECT.CDSNONEPG” or “OBJECT.RECORDSCHEDULE.DIRECT.CDSEPG”, then the srs:scheduledCDSObjectID property value shall contain the @id value for a CDS item with the upnp:recordable property value of “1”.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	NPXUH	N

NOTE: The CDS object id values used to create a recordSchedule for either the cdsNonEPG or cdsEPG record classes needs to have the upnp:recordable property set to “1”.

7.4.4.16.2

[GUIDELINE] If a UPnP AV MediaServer control point sends a SRS:CreateRecordSchedule request where the Elements input argument has the srs:class property with a value of “OBJECT.RECORDSCHEDULE.DIRECT.MANUAL”, then the srs:scheduledChannelID property value shall contain the upnp:channelID property value for a CDS item with the upnp:recordable property value of “1”.

[ATTRIBUTES]

M	C	+SR+	n/a	n/a	[60]	Y75P3	N

NOTE: The channelID values used to create a recordSchedule for the manual record class needs to have the upnp:recordable property set to "1"

7.4.4.16.3

[GUIDELINE] A UPnP AV MediaServer ScheduledRecording service shall return a success response to a SRS:CreateRecordSchedule request when the Elements input argument satisfies the following conditions:

- srs:class property has a value of "OBJECT.RECORDSCHEDULE.DIRECT.CDSNONEPG" or "OBJECT.RECORDSCHEDULE.DIRECT.CDSEPG".
- There exists a CDS object which meets the following criteria:
  - The @id property is equal to the srs:scheduledCDSObjectID property value of the request.
  - The upnp:recordable property has a value of "1".

This guideline only applies when other error conditions are not satisfied (e.g. syntax errors, resource constraints, or content recording permissions).

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	SO5WC	N

7.4.4.16.4

[GUIDELINE] A UPnP AV MediaServer ScheduledRecording service shall return a success response to a SRS:CreateRecordSchedule request when the Elements input argument satisfies the following conditions:

- srs:class property has a value of OBJECT.RECORDSCHEDULE.DIRECT.MANUAL".
- There exists a CDS object which meets the following criteria:
  - The @id property is equal to the srs:scheduledCDSObjectID property value of the request.
  - The upnp:recordable property has a value of "1".

This guideline only applies when other error conditions are not satisfied (e.g. syntax errors, resource constraints, or content recording permissions).

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	ONPRA	N

7.4.4.16.5

[GUIDELINE] If a UPnP AV MediaServer receives a SRS:CreateRecordSchedule action in which the Elements input argument is consistent with the set of allowed values returned by the SRS:GetAllowedValues action, then it may respond with 703 (Invalid Value) to indicate that the UPnP AV MediaServer is unable to create a recordSchedule using the requested input values.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[60]	NCQ8F	N

NOTE: The allowed values description returned by the SRS:GetAllowedValues action is static and does not reflect the internal real time constraints of the UPnP AV MediaServer. The UPnP AV Datastructure Template (AVDT) description cannot completely describe the semantics of a property. A UPnP AV MediaServer and a ScheduledRecording control point need to understand this limitation.

7.4.4.16.6

[GUIDELINE] If a UPnP AV MediaServer responds to a SRS:CreateRecordSchedule request with a UPnP AV error code, then the UPnP AV MediaServer should use a localized, human-readable error message in the <errorDescription> element of the SOAP response.

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[60]	QCK4X	N

NOTE: The Elements input argument is an XML document and complex. Error description is helpful to identify which property generated the error response.

7.4.4.16.7

[GUIDELINE] A UPnP AV MediaServer shall not respond to SRS:CreateRecordSchedule request with 730 (Conflict).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	DZFPW	N

NOTE: DLNA requires a UPnP AV MediaServer to always create a recordSchedule even in the event of a conflict.

7.4.4.16.8

[GUIDELINE] A UPnP AV MediaServer shall create a new recordSchedule and its associated recordTask(s) as per UPnP SRS priority model [section 2.8 of [60]].

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[60]	4W7K9	N

7.4.4.16.9

[GUIDELINE] If a UPnP AV MediaServer implements the srs:priority@orderedValue, then it shall implement the additional priority model as described in the UPnP SRS priority model for orderedPriority [section 2.8.2 of [60]].

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[60]	EHZ4X	N

7.4.4.16.10

[GUIDELINE] If a UPnP AV MediaServer creates a recordSchedule in response to a SRS:CreateRecordSchedule request, it shall set new conflict winning and losing recordTask(s) as per UPnP priority model.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	YR2SY	N

NOTE: The identification of the newly set conflict winning and losing recordTask(s) are described in guideline requirements 7.4.4.15.3 and 7.4.4.15.4

7.4.4.17 MM/SR Adjustment of Property Values for a recordSchedule or recordTask

7.4.4.17.1

[GENERAL] This defines the guidelines for a UPnP AV MediaServer when the UPnP AV MediaServer adjusts a recordSchedule or recordTask due to the device specific reasons.

7.4.4.17.2

[GUIDELINE] If a UPnP AV MediaServer creates a recordSchedule in response to the SRS:CreateRecordSchedule request, then any supported properties specified in the Elements input argument shall have the same values in the Result output argument.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[60]	OSGDD	N

NOTE: This guideline reiterates the behavior described in 2.6.7.1.3 of [60]

7.4.4.17.3

[GUIDELINE] A UPnP AV MediaServer may adjust the values of the following properties of the recordSchedule:

- srs:scheduledStartDateTime
- srs:scheduledDuration

after the UPnP AV MediaServer response to the SRS:CreateRecordSchedule request has been sent and a recordSchedule object has been created by the UPnP AV MediaServer.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[60]	6FRXA	N

NOTE: As described in 2.4.4 of [60], the SRS.StateUpdateID state variable is incremented when a recordSchedule or recordTask is modified. The reasons of the object modification are not restricted only to any ScheduledRecording service action invocations. A ScheduledRecording control point needs to monitor the changes via the SRS.StateUpdateID state variable or the SRS.LastChange evented state variable.

7.4.4.17.4

[GUIDELINE] If a UPnP AV MediaServer adjusts the srs:scheduledStateDateTime or srs:scheduledDuration property values as defined in 7.4.4.17.3, then any adjustment shall be by a maximum of one minute.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	BZOPF	N

7.4.4.17.5

[GUIDELINE] A UPnP AV MediaServer may adjust the value of the following properties of a recordTask to a value that is different from that of the parent recordSchedule object:

- srs:taskStartDateTime
- srs:taskDuration

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[60]	84MAY	N

NOTE: As described in 2.4.4 of [60], the SRS.StateUpdateID state variable is incremented when a recordSchedule or recordTask is modified. The reasons of the object modification are not restricted only to any ScheduledRecording service action invocations. A ScheduledRecording control point needs to monitor the changes via the SRS.StateUpdateID state variable or the SRS.LastChange evented state variable.

7.4.4.17.6

[GUIDELINE] If a UPnP AV MediaServer creates a recorded content based on a recordTask and succeeds the recording of the recordTask, then the actual recorded start date&time and duration of the recorded content (i.e. upnp:recordedStartDateTime and upnp:recordedDuration properties of the resulting CDS object) may be different from the actual scheduled start date&time and duration (i.e. srs:taskStartDateTime and srs:taskDuration properties) of the recordTask.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[60]	MZOBH	N

NOTE: As described in 2.2.2.22 and 2.2.2.25 of [60], the actual start date&time and duration of the recordTask may include any device specific latencies of record startup and/or teardown. A ScheduledRecording control point may not be able to retrieve the latencies from any properties of the recordTask.

7.4.4.17.7

[GUIDELINE] A UPnP AV MediaServer should start the recording at or before the value of srs:taskStartDateTime of a recordTask and should stop the recording at or after the scheduled end time of the recordTask.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[60]	KTCGR	N

NOTE: The scheduled end time of the recordTask is the result of combination of srs:taskStartDateTime and srs:taskDuration of the recordTask

7.4.4.18 MM/SR SRS:GetPropertyList Action

[GUIDELINE] If a UPnP AV MediaServer includes the value of "srs-conflict-resolution" in the <dlna:X_DLNACAP>, then it shall include the property srs:priority@orderedValue in the PropertyList output argument in response to a SRS:GetPropertyList request.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	TUOEG	N

7.4.4.19 MM/SR SRS:DeleteRecordSchedule Action

7.4.4.19.1

[GUIDELINE] If a UPnP AV MediaServer that includes the value of "srs-conflict-resolution" in the <dlna:X_DLNACAP> cannot ensure that all the intended recordTask(s) are deleted within 27 seconds in response to the SRS:DeleteRecordSchedule request then it shall respond with 720 (Cannot process the request).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	GO7PJ	N

NOTE: A UPnP control point can terminate the TCP connection after 30 seconds [7.3.2.9.5], so in order to have a consistent behavior, the UPnP AV MediaServer needs to return this error within 27 seconds.

7.4.4.19.2

[GUIDELINE] If a UPnP AV MediaServer that does not include the value of "srs-conflict-resolution" in the <dlna:X_DLNACAP> cannot ensure that all the intended recordTask(s) are deleted within 27 seconds in response to the SRS:DeleteRecordSchedule request then it should respond with 720 (Cannot process the request).

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[60]	7A2UX	N

NOTE: A UPnP control point can terminate the TCP connection after 30 seconds [7.3.2.9.5], so in order to have a consistent behavior, the UPnP AV MediaServer needs to return this error within 27 seconds.

7.4.4.20 MMSR SRS:GetRecordSchedule Action

[GUIDELINE] If a UPnP AV MediaServer cannot process a SRS:GetRecordSchedule request within 27 seconds, then it shall respond with 720 (Cannot process the request)

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	MCWTT	N

NOTE: A UPnP control point can terminate the TCP connection after 30 seconds [7.3.2.9.5], so in order to have a consistent behavior, the UPnP AV MediaServer needs to return this error within 27 seconds.

7.4.4.21 MM/SR SRS:EnableRecordSchedule Action

[GUIDELINE] If a UPnP AV MediaServer cannot ensure that the enabled recordTask(s) that resulted from SRS:EnableRecordSchedule action, and are in "ACTIVE" phase, cannot start within 60 seconds, then the SRS:EnableRecordSchedule action shall return with 720 (Cannot process the request). The UPnP AV MediaServer shall return this error within 27 seconds.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[60]	KZXAY	N

NOTE: A UPnP control point can terminate the TCP connection after 30 seconds [7.3.2.9.5], so in order to have a consistent behavior, the UPnP AV MediaServer needs to return this error within 27 seconds

7.4.4.22 MM/SR SRS:DisableRecordSchedule Action

[GUIDELINE] If a UPnP AV MediaServer, in response to SRS:DisableRecordSchedule action, cannot ensure that all the intended recordTask(s) are disabled within 27 seconds, then the SRS:DisableRecordSchedule action shall fail with a return error code 720 (cannot process the request).

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[60]	VH55K	N

NOTE: A UPnP AV MediaServer can take a long time to disable all intended recordTask(s), but the UPnP control point can terminate the TCP connection after 30 seconds [7.3.2.9.5]. So, in order to ensure a consistent UPnP AV MediaServer behavior, a status response from the UPnP AV MediaServer within 30 seconds is useful for the UPnP control point.

7.4.4.23 MM/SR SRS:GetRecordTask Action

[GUIDELINE] If a UPnP AV MediaServer cannot process SRS:GetRecordTask within 27 seconds, then the SRS:GetRecordTask action shall fail with a return error code 720 (cannot process the request).

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[60]	VUU6G	N

NOTE: A UPnP control point may terminate the TCP connection after 30 seconds [7.3.2.9.5], so in order to have a consistent behavior, the UPnP AV MediaServer needs to return this error within 27 seconds.

7.4.4.24 MM/SR SRS:EnableRecordTask Action

[GUIDELINE] If the target recordTask specified in the SRS:EnableRecordTask request is in the "ACTIVE" phase and if the UPnP AV MediaServer cannot ensure that the enabled recording start within 60 seconds, then the SRS:EnableRecordTask action shall return with error code 720 (cannot process the request).

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[60]	A2QKT	N

NOTE: A UPnP control point can terminate the TCP connection after 30 seconds [7.3.2.9.5], so in order to have a consistent behavior, the UPnP AV MediaServer needs to return this error within 27 seconds.

7.4.4.25 MM/SR SRS:ResetRecordTask Action

[GUIDELINE] If a UPnP AV MediaServer can not process SRS:ResetRecordTask request within 27 seconds, then this request shall return with error code 720 (cannot process the request).

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[60]	C787G	N

NOTE: A UPnP control point can terminate the TCP connection after 30 seconds [7.3.2.9.5], so in order to have a consistent behavior, the UPnP AV MediaServer needs to return this error within 27 seconds.

7.4.4.26 MM/SR SRS:GetRecordScheduleConflicts Action

[GUIDELINE] If a UPnP AV MediaServer cannot process SRS:GetRecordScheduleConflicts request within 27 seconds, then it shall respond with 720 (cannot process the request).

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[60]	4SOLH	N

NOTE: A UPnP control point can terminate the TCP connection after 30 seconds [7.3.2.9.5], so in order to have a consistent behavior, the UPnP AV MediaServer needs to return this error within 27 seconds.

7.4.4.27 MM/SR SRS:GetRecordTaskConflicts Action

[GUIDELINE] If a UPnP AV MediaServer cannot process SRS:GetRecordTaskConflicts request within 27 seconds, then it shall respond with 720 (cannot process the request).

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[60]	TWZDX	N

NOTE: A UPnP control point can terminate the TCP connection after 30 seconds [7.3.2.9.5], so in order to have a consistent behavior, the UPnP AV MediaServer needs to return this error within 27 seconds.

7.4.4.28 MM/SR Open-end Recording

7.4.4.28.1

[GENERAL] This defines the guidelines for a UPnP AV MediaServer when an open-end recording is requested by a ScheduledRecording control point. Duration of an “open-end” recording is determined by the UPnP AV MediaServer

7.4.4.28.2

[GUIDELINE] A UPnP AV MediaServer may implement Open-end Recording.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[60]	44YZR	N

7.4.4.28.3

[GUIDELINE] If a UPnP AV MediaServer implements Open-end Recording, then the UPnP AV MediaServer shall implement Open-end Recording for at least one of the following record classes:

- OBJECT.RECORDSCHEDULE.DIRECT.CDSNONEPG
- OBJECT.RECORDSCHEDULE.DIRECT.MANUAL

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	TV55X	N

7.4.4.28.4

[GUIDELINE] The UPnP AV MediaServer shall not implement Open-end Recording for record classes other than the following:

- OBJECT.RECORDSCHEDULE.DIRECT.CDSNONEPG
- OBJECT.RECORDSCHEDULE.DIRECT.MANUAL

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	D97BI	N

7.4.4.28.5

[GUIDELINE] If a UPnP AV MediaServer implements Open-end Recording, it shall include the value of dlna:openDuration in response to the SRS:GetPropertyList action when the DataTypeID input argument is the value A_ARG_TYPE_RecordScheduleParts, A_ARG_TYPE_RecordSchedule or A_ARG_TYPE_RecordTask.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	KJSNG	N

7.4.4.28.6

[GUIDELINE] If a UPnP AV endpoint includes a dlna:openDuration element in A_ARG_TYPE_RecordScheduleParts, A_ARG_TYPE_RecordSchedule and A_ARG_TYPE_RecordTask, then it shall use the following semantics and syntax.

Table 37 — dlna:openDuration Property Type and Multi Value

Property Name	Property Type	Multiple Value
dlna:openDuration	xsd:boolean	No

The value of dlna:openDuration element shall be one of the followings. The default value is "0".

- "1" when the recording is Open-end recording.
- "0" when the recording is not Open-end recording.

The prefix for dlna:openDuration shall be "dlna" and the namespace shall be "urn:schemas-dlna-org:metadata-1-0/".

[ATTRIBUTES]

M	A	DMS +SR+	M-DMS	n/a	[60]	IDGSQ	N

7.4.4.28.7

[GUIDELINE] If a UPnP AV MediaServer implements Open-end Recording, the UPnP AV MediaServer shall respond to the SRS:GetAllowedValues action for the dlna:openDuration property with the PropertyInfo output argument which contains an AVDT description for dlna:openDuration that is consistent with guidelines 7.4.4.28.3, 7.4.4.28.4, and 7.4.4.28.6.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	YU6OF	N

NOTE: In case that the Open-end Recording is available for cdsNonEPG and manual class recordings, the concrete examples of <field> element in the AVDT description are as follows:

- When the DataTypeID input argument is the value A_ARG_TYPE_RecordScheduleParts,

```
<field>
    <name>dlna:openDuration</name>
    <dataType>xsd:boolean</dataType>
    <allowedValueDescriptor>
        <defaultValue>0</defaultValue>
        <dependentField>
            <name>srs:class</name>
            <valueList>
                <value>OBJECT.RECORDSCHEDULE.DIRECT.CDSNONEPG</value>
                <value>OBJECT.RECORDSCHEDULE.DIRECT.MANUAL</value>
            </valueList>
        </dependentField>
        <allowedValueList>
            <allowedValue>0</allowedValue>
            <allowedValue>1</allowedValue>
        </allowedValueList>
    </allowedValueDescriptor>
</field>
```

- When the DataTypeID input argument is the value A_ARG_TYPE_RecordSchedule,

```
<field>
    <name>dlna:openDuration</name>
    <dataType>xsd:boolean</dataType>
    <allowedValueDescriptor>
        <minCount>1</minCount>
        <dependentField>
            <name>srs:class</name>
            <valueList>
                <value>OBJECT.RECORDSCHEDULE.DIRECT.CDSNONEPG</value>
                <value>OBJECT.RECORDSCHEDULE.DIRECT.MANUAL</value>
            </valueList>
        </dependentField>
        <allowedValueList>
            <allowedValue>0</allowedValue>
            <allowedValue>1</allowedValue>
        </allowedValueList>
    </allowedValueDescriptor>
</field>
```

- When the DataTypeID input argument is the value A_ARG_TYPE_RecordTask,

```
<field>
    <name>dlna:openDuration</name>
    <dataType>xsd:boolean</dataType>
    <allowedValueDescriptor>
        <minCount>1</minCount>
        <allowedValueList>
            <allowedValue>0</allowedValue>
            <allowedValue>1</allowedValue>
        </allowedValueList>
    </allowedValueDescriptor>
</field>
```

7.4.4.28.8

[Guideline] If a UPnP AV MediaServer control point creates a recordSchedule in which the dlna:openDuration property has a value of “1” , then it shall specify “P00:00:00” as the value of the srs:scheduledDuration element in the request of SRS:CreateRecordSchedule action.

[Attributes]

M	A	+SR+	n/a	n/a	[60]	KKQRH	N

NOTE: [60] requires a ScheduledRecording control point to always specify a value for the srs:scheduledDuration property when creating a cdsNonEPG or manual record class recordSchedule. This guideline defines the value to be used in the request when an Open-end Recording is desired.

7.4.4.28.9

[Guideline] If a UPnP AV MediaServer implements Open-end Recording, the UPnP AV MediaServer shall return a success response to a SRS:CreateRecordSchedule request when the Elements input argument satisfies the following conditions:

- srs:class property has a value of "OBJECT.RECORDSCHEDULE.DIRECT.CDSNONEPG" or "OBJECT.RECORDSCHEDULE.DIRECT.MANUAL".
- srs:scheduledDuration has a value of “P00:00:00”.
- dlna:openDuration property has a value of “1”.

This guideline only applies when error conditions are not satisfied (e.g. syntax errors, resource constraints, or content recording permissions).

[Attributes]

M	A	DMS	M-DMS	n/a	[60]	VCTAV	N

7.4.4.28.10

[Guideline] If a UPnP AV MediaServer creates a recordTask based on a recordSchedule in which the dlna:openDuration property has a value of “1”, the value of srs:taskDuration property of the recordTask shall be “P00:00:00” until the UPnP AV MediaServer determines the exact duration.

[Attributes]

M	A	DMS	M-DMS	n/a	[60]	JPYH6	N

NOTE: <Delete entire paragraph if there is no note.>

7.4.4.29 MM/SR Media Format Specified Recording

7.4.4.29.1

[General] This contains guidelines which define the optional media format specified recording scheme using a DLNA Media Format Profile

7.4.4.29.2

[Guideline] A UPnP AV MediaServer may accept dlna:desiredPN as an element in A_ARG_TYPE_RecordScheduleParts for media format specified recording. The semantics and syntax of dlna:desiredPN element is defined in 7.4.4.29.4.

[Attributes]

O	A	DMS	M-DMS	n/a	[60]	4L8AD	N

NOTE: DLNA introduces optional media format specified recording mechanism in this requirement. If a UPnP AV MediaServer control point uses this mechanism, then it can improve the playback of the recorded contents with its supported DLNA Media Format Profile

7.4.4.29.3

[GUIDELINE] If a UPnP AV MediaServer accepts dlna:desiredPN element in A_ARG_TYPE_RecordScheduleParts, then it shall implement dlna:PN as an element in A_ARG_TYPE_RecordTask state variable for media format specified recording. The semantics and syntax of dlna:PN element is defined in 7.4.4.29.6.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	LESSU	N

7.4.4.29.4

[GUIDELINE] If a UPnP AV MediaServer or UPnP AV MediaServer control point includes a dlna:desiredPN element in A_ARG_TYPE_RecordScheduleParts, A_ARG_TYPE_RecordSchedule or A_ARG_TYPE_RecordTask, then it shall use the following semantics and syntax.

The syntax definition of dlna:desiredPN element shall be as follows:

Table 38 — dlna:desiredPN Property Type and Multi Value

Property Name	Property Type	Multiple Value
dlna:desiredPN	CSV(String)	No

The value of dlna:desiredPN element shall be one of the followings.

- One DLNA Media Format Profile
- “AUTO” (This meaning is that the UPnP AV MediaServer is free to use any DLNA Media Format Profiles.)
- One CSV list which includes “AUTO” and/or DLNA Media Format Profile(s)

If the value of dlna:desiredPN is a CSV list, then the DLNA Media Format Profiles in the CSV list shall be ordered in the preferred formats for the recording, where the first format is the most preferred. If “AUTO” is included in the list, it shall appear as the last value in the list and indicates that if none of the preceding values are available, then the UPnP AV MediaServer is free to use any DLNA Media Format Profiles to maximize the probability that the recording actually takes place.

In addition, the DLNA Media Format Profile shall omit the DLNA Link Protection prefix, e.g. “DTCP_” for DTCP-IP and “WMDRM_” for WMDRM-ND.

The prefix for dlna:desiredPN shall be "dlna" and the namespace shall be “urn:schemas-dlna-org:metadata-1-0/”.

[ATTRIBUTES]

M	A	DMS +SR+	M-DMS	n/a	[60]	T9VRU	N

NOTE: Example usages for dlna:desiredPN would be as follows:

- `<dlna:desiredPN>MPEG_TS_JP_T</dlna:desiredPN>`
- `<dlna:desiredPN>AUTO</dlna:desiredPN>`
- `<dlna:desiredPN> MPEG_TS_JP_T,MPEG_PS_NTSC,AUTO</dlna:desiredPN>`

7.4.4.29.5

[GUIDELINE] A UPnP AV MediaServer shall accept the value “AUTO” for the dlna:desiredPN element.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	XYGA5	N

NOTE: The value and semantics for “AUTO” needs to be implemented by a UPnP AV MediaServer.

7.4.4.29.6

[GUIDELINE] If a UPnP AV MediaServer includes a dlna:PN element in A_ARG_TYPE_RecordTask, then it shall use the following semantics and syntax.

The syntax definition of dlna:PN element shall be as follows:

Table 39 — dlna:PN Property Type and Multi Value

Property Name	Property Type	Multiple Value
dlna:PN	String	No

The value of dlna:PN element shall be one DLNA Media Format Profile which will be used the recording. When the recordTask is in the “IDLE” phase, this property shall contain a best-known estimate of DLNA Media Format Profile for the recording. When the recordTask is in the “ACTIVE” or “DONE” phase, this property shall contain one of the DLNA Media Format Profiles supported by the UPnP AV MediaServer for the actual recording.

In addition, the property value shall not include any profile that has the DLNA Link Protection prefix, e.g. “DTCP_” for DTCP-IP and “WMDRM_” for WMDRM-ND.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	9K9SU	N

NOTE: “AUTO” is not allowed for the value of this element. An example usage for the dlna:PN element would be as follows:

- `<dlna:PN>MPEG_PS_NTSC</dlna:PN>`

7.4.4.29.7

[GUIDELINE] If a UPnP AV MediaServer control point does not specify DLNA Media Format Profile(s), then it shall omit dlna:desiredPN eleIment or specify only “AUTO” as the value of the dlna:desiredPN element in the request of SRS:CreateRecordSchedule action.

[ATTRIBUTES]

M	A	+SR+	n/a	n/a	[60]	FZIUX	N

NOTE: If a UPnP AV MediaServer control point omit dlna:desiredPN element or specify only “AUTO” as the value of the dlna:desiredPN element, then the UPnP AV MediaServer is free to select a DLNA Media Format Profile for the recording.

7.4.4.29.8

[GUIDELINE] If a UPnP AV MediaServer control point requests to record a content using one DLNA Media Format Profile from the CSV list of the dlna:desiredPN element, then it shall not include “AUTO” in the CSV list of dlna:desiredPN element on the request of SRS:CreateRecordSchedule action.

[ATTRIBUTES]

M	A	+SR+	n/a	n/a	[60]	JUP3C	N

NOTE: This guideline enable a UPnP AV MediaServer control point to request to record a content using the DLNA Media Format Profile which it specified on the request of SRS:CreateRecordSchedule action.

7.4.4.29.9

[Guideline] If a UPnP AV MediaServer accepts dlna:desiredPN element, then it shall create one or more recordTask(s) with dlna:PN element regardless of whether the UPnP AV MediaServer control point specifies the dlna:desiredPN element or not.

[Attributes]

M	A	DMS	M-DMS	n/a	[60]	ZI2N7	N

NOTE: A UPnP AV MediaServer control point can retrieve the selected DLNA Media Format Profile for the recording via the response of SRS:BrowseRecordTasks action which has the dlna:PN element in advance.

7.4.4.29.10

[Guideline] If a UPnP AV MediaServer accepts dlna:desiredPN element, then it shall adhere the following rules for selecting a DLNA Media Format Profile for dlna:PN element and a level of recording quality for srs:recordQuality element:

- If a UPnP AV MediaServer control point omits dlna:desiredPN element or specifies only “AUTO” in the dlna:desiredPN element, then the UPnP AV MediaServer shall select the acceptable and most preferable level of recording quality from CSV list of srs:desiredRecordQuality element and the corresponding DLNA Media Format Profile.
- If there are some acceptable combinations that the selected value of dlna:desiredPN is not “AUTO”, then the UPnP AV MediaServer shall select the acceptable and most preferable combination of level of recording quality from CSV list of srs:desiredRecordQuality element and DLNA Media Format Profile from CSV list of dlna:desiredPN element with the acceptable and most preferable DLNA Media Format Profile.
- If a UPnP AV MediaServer control point didn’t include “AUTO” in dlna:desiredPN element and there is no acceptable combination, then the UPnP AV MediaServer shall select the acceptable and most preferable specified DLNA Media Format Profile from CSV list of dlna:desiredPN element and the corresponding level of recording quality from CSV list of srs:desiredRecordQuality.
- If a UPnP AV MediaServer control point omits srs:desiredRecordQuality element or specifies only “AUTO” in the srs:desiredRecordQuality element, then the UPnP AV MediaServer shall select the acceptable and most preferable specified DLNA Media Format Profile from CSV list of dlna:desiredPN element and the corresponding level of recording quality.

[Attributes]

M	A	DMS	M-DMS	n/a	[60]	73XFC	N

NOTE: This requirement for a UPnP AV MediaServer provides selection criteria of recording quality and DLNA Media Format Profile that a UPnP AV MediaServer control point specified.

In this guideline, there are four criteria according to the following combination of specified recording qualities and DLNA Media Format Profiles;

- Only recording qualities are specified.
- Both recording qualities and DLNA Media Format Profiles are specified
  - A UPnP AV MediaServer will be able to record using one of the requested recording qualities and one of the DLNA Media Format Profiles.
  - A UPnP AV MediaServer will not be able to record using one of the requested recording qualities and one of DLNA Media Format Profiles.
- Only DLNA Media Format Profiles are specified.

If a UPnP AV MediaServer control point specifies DLNA Media Format Profiles, then the DLNA Media Format Profiles take precedence of the recording qualities since the control point wants to play back the recorded content with the specified DLNA Media Format Profiles. The timing of the selection is vendor dependent.

7.4.4.29.11

[GUIDELINE] If a UPnP AV MediaServer implements dlna:desiredPN property, it shall include the value of dlna:desiredPN in response to the SRS:GetPropertyList action when the DataTypeID input argument is the value A_ARG_TYPE_RecordScheduleParts, A_ARG_TYPE_RecordSchedule or A_ARG_TYPE_RecordTask.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	36UYV	N

7.4.4.29.12

[GUIDELINE] If a UPnP AV MediaServer implements dlna:PN property, it shall include the value of dlna:PN in response to the SRS:GetPropertyList action when the DataTypeID input argument is the value A_ARG_TYPE_RecordTask.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	HWUAU	N

7.4.4.29.13

[GUIDELINE] If a UPnP AV MediaServer implements dlna:desiredPN property, then it shall respond to the SRS:GetAllowedValues action for the dlna:desiredPN property with the PropertyInfo output argument which contains an AVDT description for dlna:desiredPN that is consistent with guideline 7.4.4.29.4. The allowed values in the AVDT description shall be listed in order of quality from highest quality to lowest. The value “AUTO” shall always be present and appear as the last item in the list.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	TKB45	N

NOTE: DLNA does not define the ordering of quality between DLNA Media Format Profiles. The specific ordering is vendor-dependent, and is communicated through the AVDT description. For example, if the vendor defines the ordering of the recording quality for DLNA Media Format Profiles to be MPEG_TS_JP_T > MPEG_PS_NTSC and the DataTypeID input argument value is A_ARG_TYPE_RecordScheduleParts, the concrete example of the AVDT description that adheres to this requirement is as follows;

```
<?xml version="1.0" encoding="UTF-8"?>
<AVDT
   xmlns:xsd=http://www.w3.org/2001/XMLSchema
   xmlns="urn:schemas-upnp-org:av:avdt"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:av:avdt
   http://www.upnp.org/schemas/av/avdt-v1-20060531.xsd">
   <contextID>
      uuid:device-UUID::urn:schemas-upnp-org:service:ScheduledRecording:1
   </contextID>
   <dataStructType>A_ARG_TYPE_RecordScheduleParts</dataStructType>
   <fieldTable>
      <field>
         <name>dlna:desiredPN</name>
         <dataType csv="xsd:string" maxSize="256">xsd:string</dataType>
         <maxListSizeTotal>UNBOUNDED</maxListSizeTotal>
         <allowedValueDescriptor>
            <defaultValue>AUTO</defaultValue>
            <allowedValueList>
               <allowedValue>MPEG_TS_JP_T</allowedValue>
               <allowedValue>MPEG_PS_NTSC</allowedValue>
               <allowedValue>AUTO</allowedValue>
            </allowedValueList>
```

```
         </allowedValueDescriptor>
      </field>
   </fieldTable>
</AVDT>
```

7.4.4.29.14

[Guideline] If a UPnP AV MediaServer implements dlna:PN property, then it shall respond to the SRS:GetAllowedValues action for the dlna:PN property with the PropertyInfo output argument which contains an AVDT description for dlna:PN that is consistent with guideline requirement 7.4.4.29.6. The allowed values in the AVDT description shall be listed in order of quality from highest quality to lowest.

[Attributes]

M	A	DMS	M-DMS	n/a	[60]	ORUDT	N

NOTE: In case that the ordering of the recording quality of DLNA Media Format Profile is MPEG_TS_JP_T > MPEG_PS_NTSC, the concrete example of the AVDT description that adheres to this requirement is as follows;

```
<?xml version="1.0" encoding="UTF-8"?>
<AVDT
   xmlns:xsd=http://www.w3.org/2001/XMLSchema
   xmlns="urn:schemas-upnp-org:av:avdt"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:av:avdt
   http://www.upnp.org/schemas/av/avdt-v1-20060531.xsd">
   <contextID>
      uuid:device-UUID::urn:schemas-upnp-org:service:ScheduledRecording:1
   </contextID>
   <dataStructType>A_ARG_TYPE_RecordTask</dataStructType>
   <fieldTable>
      <field>
         <name>dlna:PN</name>
         <dataType maxSize="256">xsd:string</dataType>
         <allowedValueDescriptor>
            <minCount>1</minCount>
            <allowedValueList>
               <allowedValue>MPEG_TS_JP_T</allowedValue>
               <allowedValue>MPEG_PS_NTSC</allowedValue>
            </allowedValueList>
         </allowedValueDescriptor>
      </field>
   </fieldTable>
</AVDT>
```

7.4.4.30 EPG, SRS, and CDS Object Lifespan Guidelines

7.4.4.30.1 General

7.4.4.30 defines the guidelines for a UPnP AV MediaServer when implementing the optional EPG Server Device Option and the Scheduled Recording Device Option. It defines the “lifespan” requirements for recordSchedule and recordTask of cdsEPG record class, CDS objects generated by a recordTask, and EPG Program Items.

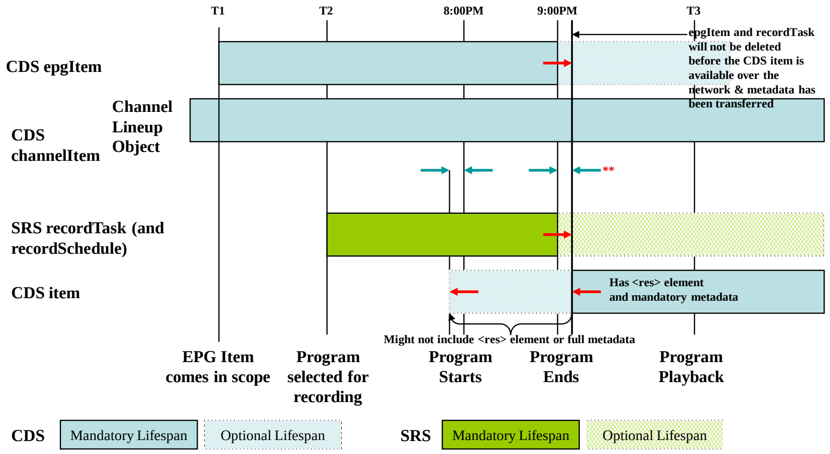


Figure 30 — CDS and SRS Object Lifetimes

7.4.4.30.2 MM/SR EPG Program Items Lifetime

7.4.4.30.2.1

[GUIDELINE] An EPG Program Item shall exist until the upnp:scheduledEndTime.

[ATTRIBUTES]

M	A	DMS	M-MDS	n/a	[46]	VRL2U	N

NOTE: An EPG Program Item could be assumed to exist before the upnp:scheduledStartTime.

7.4.4.30.2.2

[GUIDELINE] If an EPG Program Item have a upnp:channelID property value which refers to a CDS channel item, then the CDS channel item shall exist in a Tuner container

[ATTRIBUTES]

M	A	DMS	M-MDS	n/a	[46]	SPUWB	N

NOTE: <Delete entire paragraph if there is no note.>

7.4.4.30.3 MM/SR recordTask Lifespan

7.4.4.30.3.1

[GENERAL] 7.4.4.30.3 defines the guidelines for a UPnP AV MediaServer when implementing the Scheduled Recording Device Option. It defines the "lifespan" requirements for recordTask.

7.4.4.30.3.2

[GUIDELINE] For purposes of these guidelines, if the UPnP AV MediaServer does not expose the recorded content in the CDS, then an SRS recordTask is considered "Completed" when the srs:taskState property is set to one of the following values:

- DONE.FULL
- DONE.PARTIAL
- DONE.EMPTY

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	9SQV4	N

NOTE: This guideline defines the meaning of "Completed" for an SRS recordTask. The completion of a recordTask is a key event for guidelines defining the lifespan of EPG, SRS, and CDS objects.

7.4.4.30.3.3

[GUIDELINE] For purposes of these guidelines, if the UPnP AV MediaServer exposes the recorded content in the CDS, an SRS recordTask is considered "Completed" when one of the following two conditions are met:

- the srs:taskState property value is set to DONE.EMPTY
- the srs:taskState property value is set to DONE.FULL or DONE.PARTIAL and
  - the CDS object associated with the recordTask exists, and
  - res property with a URI for streaming playback is available

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	U8MPL	N

NOTE: This guideline defines the meaning of "Completed" for an SRS recordTask. The completion of a recordTask is a key event for guidelines defining the lifespan of EPG, SRS, and CDS objects.

7.4.4.30.3.4

[GUIDELINE] A UPnP AV MediaServer shall retain the recordTask object that is in "ACTIVE" phase until the scheduled endtime of the recordTask, regardless of whether errors, conflicts, or other conditions that allow the recordTask to complete successfully or not.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	4OZNM	N

NOTE: UPnP AV MediaServer control points need to retrieve the current status of a recordTask via the SRS:BrowseRecordTasks action during the expected recording time.

7.4.4.30.3.5

[GUIDELINE] When a UPnP AV MediaServer receives an SRS:DeleteRecordTask action on a recordTask, it shall delete the recordTask object. This guideline overrides the guideline 7.4.4.30.3.4 for the SRS:DeleteRecordTask case.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	7OTK7	N

7.4.4.30.3.6

[GUIDELINE] A UPnP AV MediaServer shall retain a recordSchedule object until all its associated recordTask objects have been deleted.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[60]	84KL6	N

NOTE: A recordTask can only exist with a parent recordSchedule and never orphaned. The recordScheduleID property contains the value of the @id property of the recordSchedule that generated the recordTask. A SRS:DeleteRecordSchedule action on a recordSchedule object with one or more associated recordTask objects in the “ACTIVE” phase will generate error 705.

7.4.4.30.3.7

[GUIDELINE] A UPnP AV MediaServer should retain a recordTask after the associated recording is completed, where “completed” is defined in 7.4.4.30.3.2 and 7.4.4.30.3.3.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[60]	CT53F	N

NOTE: When a recordTask’s end time is reached (that is: the content is no longer available) or a fatal error is detected, the associated recording finishes. If the UPnP AV MediaServer retains a recordTask after the associated recording finishes, the srs:taskState@phase attribute of the recordTask has the value of “DONE”. A recordTask in which the srs:taskState@phase attribute has the value of “DONE” may have information about a recorded content and/or error(s). Some UPnP AV MediaServers may not be able to retain a recordTask in which the srs:taskState@phase attribute has the value of “DONE” due to the device specific reasons. Furthermore some UPnP AV MediaServers may not be able to retain a recordSchedule in which the srs:scheduleState property has the value of “COMPLETED” due to the device specific reasons.

7.4.4.30.3.8

[GUIDELINE] If a UPnP AV MediaServer always retains a recordTask which has the srs:taskState@phase attribute with a value of “DONE”, then it shall use the <dlna:X_DLNACAP> element (as a child of the <device> element that represents the UPnP AV MediaServer) in the device description document and include the CapabilityID “srs-rt-done-retained” in the element’s comma-separated value list. Conversely, if a UPnP AV MediaServer does not always retain a recordTask which has the srs:taskState@phase attribute with a value of “DONE”, it shall not include the Capability ID “srs-rt-done-retained” in the <dlna:X_DLNACAP> element’s comma-separated value list in the device description document.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	EXKO4	N

NOTE: A UPnP MediaServer control point can understand that it may be able to retrieve the result of a recordTask in the UPnP AV MediaServer.

7.4.4.30.3.9

[GUIDELINE] If a UPnP AV MediaServer can always indicate the unsuccessful completion of a recordTask by retaining the recordTask after the srs:scheduledEndDateTime, then it shall use the <dlna:X_DLNACAP> element 7.3.2.35.1 in the device description document and include the Capability ID “srs-rt-can-report-unsuccessful-completion” in the element’s comma-separated value list..

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[60]	8NYC7	N

7.4.4.30.3.10

[Guideline] If a UPnP AV MediaServer always retains a recordTask as per 7.4.4.30.3.8 or 7.4.4.30.3.9 then it shall use the <dlna:X_DLNACAP> element (as a child of the <device> element that represents the UPnP AV MediaServer) in the device description document and include the capability ID and its value in the element's comma-separated value list in the format "srs-rt-retention-period-duration" where "srs-rt-retention-period-" is a literal string. The duration portion shall be a ui4 value or "infinity", indicating the number of seconds the recordTask is retained by the device after the recording is completed or aborted. The duration portion cannot be zero.

More formally, the syntax of the capability ID is defined in Table 40.

Table 40 — Capability ID Syntax

Capability ID	Description
srs-rt-retention-period-duration	The UPnP AV MediaServer supports retaining recordTask for a specific duration after "ACTIVE" state
• srs- retention- capability-id = "srs-rt-retention-period-" duration • duration = <ui4 value> | "infinity" • The "srs-rt-retention-period-" is a literal. The duration shall have a non-zero ui4 value or the literal "infinity". If it is ui4 value, then it represents the number of seconds the recordTask will be retained by the device. The literal "infinity" represents that the device will retain the recordTask until it is explicitly deleted.	

[Attributes]

M	A	DMS	M-DMS	n/a	[60]	CJ6IL	N

NOTE: AV MediaServer devices use the <dlna:X_DLNACAP> element to indicate to control points their recordTask retention periods. See guideline 7.3.2.35.1 for the formal syntax of the <dlna:X_DLNACAP> element. Sample descriptions are given below:

- ```
  <dlna:X_DLNACAP xmlns:dlna="urn:schemas-dlna-org:device-1-0">
  srs-rt-retention-period-100
  </dlna:X_DLNACAP>
  ```
- ```
  <dlna:X_DLNACAP xmlns:dlna="urn:schemas-dlna-org:device-1-0">
  srs-rt-retention-period-128
  </dlna:X_DLNACAP>
  ```
- ```
  <dlna:X_DLNACAP xmlns:dlna="urn:schemas-dlna-org:device-1-0">
  srs-rt-retention-period-infinity
  </dlna:X_DLNACAP>
  ```

7.4.4.30.3.11

[Guideline] If a UPnP AV MediaServer returns a <Feature> element in response to the CDS:GetFeatureList request with the Feature@name attribute set to a value of "DLNA.ORG_SRS_CONTENT", then it should implement srs:recordedCDSObjectID property for recordTask(s).

[Attributes]

S	C	DMS	M-DMS	n/a	[60]	6VDID	N

NOTE: This guideline is a clarification that a recordTask is able to have a srs:recordedCDSObjectID property value
  ```

7.4.4.30.3.12

[GUIDELINE] If a UPnP AV MediaServer returns a <Feature> element in response to the CDS:GetFeatureList request with the Feature@name attribute set to a value of "DLNA.ORG_SRS_CONTENT", and the srs:recordedCDSObjectID property of a recordTask exists, then the srs:recordedCDSObjectID property shall have the value of the @id property of a CDS object which exists in the CDS and represents the content recorded by the recordTask

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[60]	CHAAK	N

7.4.4.30.3.13

[GUIDELINE] If a UPnP AV MediaServer returns a <Feature> element in response to the CDS:GetFeatureList request with the Feature@name attribute set to a value of "DLNA.ORG_SRS_CONTENT", and the srs:recordedCDSObjectID property of a recordTask exists, then the value of srs:recordedCDSObjectID property of a recordTask should be set in 27 seconds after the recordTask is created.

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[60]	MDYI4	N

NOTE: It is recommended to create the srs:recordedCDSObjectID as early as possible (possibly before the recording starts) so that it can be tracked. However it is possible for some implementations to set this property after the recording is complete.

7.4.4.30.3.14

[GUIDELINE] A UPnP AV MediaServer may change the values of the srs:taskStartDateTime and/or srs:taskDuration properties of a recordTask due to the updates of the associated program information in the device internal program information source in the case of cdsEPG record class.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[60]	PT2EL	N

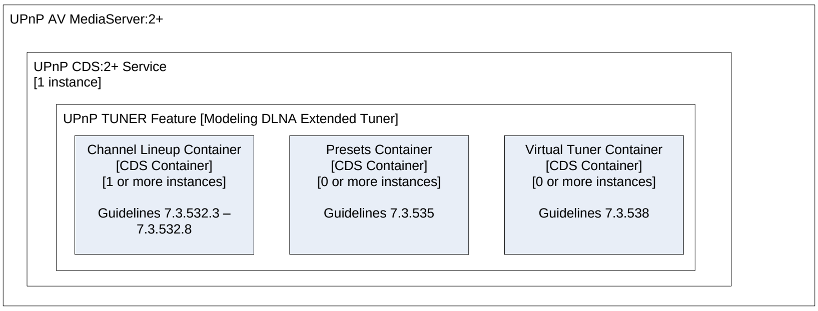
NOTE: A ScheduledRecording control point may be able to know the occurrence of changes via the event notifications

7.4.5 Extended Tuner Media Management Guidelines

7.4.5.1 General

The Basic Tuner guidelines were based on initial version of the DLNA Interoperability Guidelines and were not based on the UPnP TUNER feature. DLNA is aligning devices that implement to UPnP AV MediaServer:2 and above to implement the UPnP TUNER feature.

The UPnP TUNER feature allows UPnP AV MediaServers to implement one or more tuner containers and each of these container's CDS object ids as listed in the UPnP TUNER feature. In the DLNA Extended Tuner these tuner containers are modeled as Channel Lineup Containers, Presets Containers, and Virtual Tuner Containers. An Extended Tuner will contain at least one Channel Lineup Container. The Presets Container and Virtual Tuner Containers are optional. The figure below illustrates an Extended Tuner and its containers.



7.4.5.2 MM/ET Extended Tuner Guidelines

7.4.5.2.1

[GUIDELINE] A UPnP AV MediaServer that implements to UPnP AVv2 or higher may implement an Extended Tuner.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[49]	UCVFO	N

NOTE: Implementation of the DLNA Extended Tuner guidelines is optional in DLNA.

7.4.5.2.2

[GUIDELINE] If a UPnP AV MediaServer implements the EPG Server Device Option, as defined in 7.4.6.2.3, by including a <Feature> element with the Feature@name attribute equal to “EPG” in the FeatureList output argument in response to the CDS:GetFeatureList action and the ScheduledRecording Device Option, then it shall implement an Extended Tuner.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[49]	MNH6T	N

NOTE: Implementation of the DLNA Extended Tuner guidelines is mandated when implementing both the ScheduedRecording Device Option and EPG Server Device Options in DLNA.

7.4.5.2.3

[GUIDELINE] If a UPnP AV MediaServer implements an Extended Tuner, then it shall implement to UPnP AVv2 or higher (i.e. UPnP AV MediaServer:2 or higher) and conform to guidelines 7.4.5.3 – 7.4.5.9.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[49]	JAUDH	N

NOTE: Tuners implemented on a UPnP AV MediaServer:2 or higher are recommended to implement the Extended Tuner.

7.4.5.2.4

[GUIDELINE] A UPnP AV MediaServer control point that interacts with a UPnP AV MediaServer tuner, shall be able to browse CDS items for both the DLNA Basic Tuner (defined in guidelines (7.4.1.4.15 through 7.4.1.4.22) and the DLNA Extended Tuner (defined in guidelines 7.4.5.3 through 7.4.5.5 and 7.4.5.7 through 7.4.5.8.

[ATTRIBUTES]

M	A	DMP DMC +SR+	M-DMP M-DMC	n/a	[49]	DC6OF	N

NOTE: UPnP AV MediaServers are allowed to select which DLNA Tuner (Basic or Extended) to implement. To maintain a minimum level of interoperability a UPnP AV MediaServer control point needs to work with both of the DLNA Tuner implementations. UPnP AV MediaServer control points are not required to interact with Presets Containers (7.4.5.6) or Virtual Tuners Containers (7.4.5.9)

7.4.5.3 MM/ET Extended Tuner Common Guidelines

7.4.5.3.1

[GENERAL] These common guideline requirements for an Extended Tuner define the relationship between the DLNA Extended Tuner and the UPnP AV MediaServer ContentDirectory service TUNER feature. They define Channel Lineup Containers which expose the available channels to UPnP AV MediaServer control points. In addition, they define the common guidelines for CDS items (i.e. Non-Streamable Channel Objects and Streamable Channel Objects) that represent available channels. Non-Streamable Channel Objects in a Channel Lineup Container would be used for Extended Tuners that do not stream their content over the network, but provides CDS object IDs for internal Tuner resources that can be used to setup a Scheduled Recording with identifying channel information (cdsNonEPG record class). Optionally, Non-Streamable Channel Objects in a Channel Lineup Container could be used for Extended Tuners that implement Virtual Tuner Objects in a Virtual Tuner Container to stream selected channel content over the network through a single connection. The figure below illustrates a Channel Lineup Container.

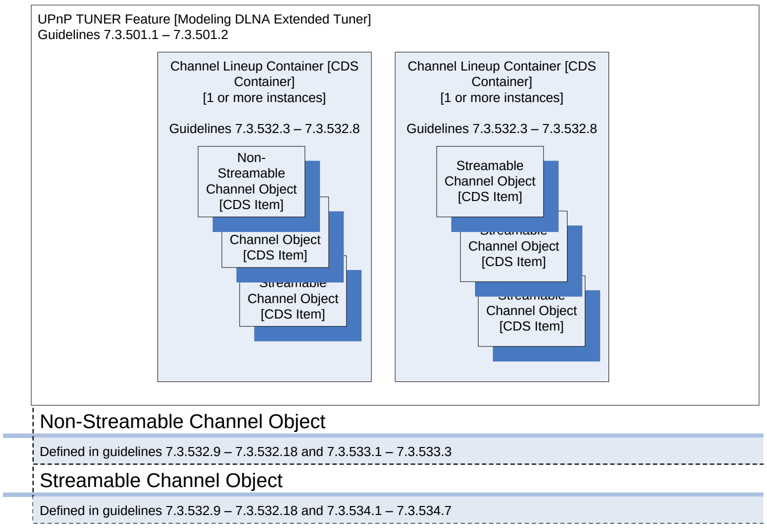


Figure 31 — Modeling DLNA Extended Tuner

7.4.5.3.2

[GUIDELINE] A UPnP AV MediaServer shall conform to all of the requirements of the TUNER Feature defined in Appendix E.2 of [46].

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	BOAGO	N

NOTE: The DLNA Tuner requirements are built upon the UPnP TUNER feature defined in the ContentDirectory service of AVv2 and higher.

7.4.5.3.3

[Guideline] A UPnP AV MediaServer shall return a <Feature> element indicating the TUNER feature is implemented in response to the CDS:GetFeatureList action.

[Attributes]

M	C	DMS	M-DMS	n/a	[46]	RMIPF	N

NOTE: The TUNER feature allows a UPnP AV MediaServer control point to determine whether a device supports the Extended Tuner functionality, and where the Channel Lineup Containers are located in the larger ContentDirectory service Container hierarchy.

7.4.5.3.4

[Guideline] A UPnP AV MediaServer shall expose at least one Channel Lineup Container in the ContentDirectory service which conforms to the requirements for Channel Lineup Containers as defined in 7.4.5.3.5 – 7.4.5.3.10.

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	7TB6O	N

NOTE: Implementation of the DLNA Extended Tuner guidelines requires at least one Channel Lineup Container be exposed by a UPnP AV MediaServer.

7.4.5.3.5

[Guideline] A UPnP AV MediaServer shall set the upnp:class property value of a Channel Lineup Container to one of the following:

- object.container.channelGroup
- object.container.channelGroup.audioChannelGroup
- object.container.channelGroup.videoChannelGroup
- class derived from any of the above classes

[Attributes]

M	R	DMS	M-DMS	n/a	[46]	PI9Z2	N

NOTE: [46] Requires channel containers to have the upnp:class of object.container.channelGroup or any of its derived classes, which gives more precise guidance on upnp:class usage than the requirement in the guideline 7.4.1.4.15.3 for a Basic Tuner.

[46] Appendix C.2.2.5 specifies that object.container.channelGroup.audioChannelGroup only contains object.item.audioItem.audioBroadcast items and object.container.channelGroup.videoChannelGroup only contains object.item.videoItem.videoBroadcast items.

7.4.5.3.6

[Guideline] A UPnP AV MediaServer with a Channel Lineup Container that contains both audioBroadcast and videoBroadcast items shall set the upnp:class property value to object.container.channelGroup or a derived class of object.container.channelGroup.

[Attributes]

M	C	DMS	M-DMS	n/a	[46]	RBSWW	N

NOTE: [46] provides this guidance.

7.4.5.3.7

[GUIDELINE] A UPnP AV MediaServer Channel Lineup Container should include a value for the upnp:channelGroupName property. The value represents the user-friendly name of the Channel Lineup Container.

[ATTRIBUTES]

S	R	DMS	M-DMS	n/a	[46]	CPID4	N

NOTE: A channel group defines a group of channels. A device that has multiple tuners can provide multiple channel groups. Moreover, a physical tuner device can provide multiple channel groups (for example, a set-top-box that contains a single tuner but supports three different input connections: terrestrial, cable, and satellite).

7.4.5.3.8

[GUIDELINE] A UPnP AV MediaServer Channel Lineup Container that includes a value for the upnp:channelGroupName property shall include the upnp:channelGroupName@id property. The upnp:channelGroupName@id property contains the ID of a channel group to differentiate it from other channel groups implemented in a ContentDirectory service. The format of the upnp:channelGroupName@id property is as follows:

```
<ICANN registered domain> "_" <channel group id defined in the domain>
```

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	G9RUP	N

NOTE: Vendors are allowed to utilize the DLNA.ORG_DefaultChannelGroup if there is no external registered domain reference available. An example for a valid value for the upnp:channelGroupName@id property is "megaserviceprovider.com_DigitalSatellite". Note that the datatype defined in [46] for the upnp:channelGroupName@id property is xsd:string, so any characters allowed by xsd:string can be used within <channel group id defined in this domain>. For example, "DLNA.ORG_Any valid characters-+&23" would be valid syntax through not semantically useful.

7.4.5.3.9

[GUIDELINE] A UPnP AV MediaServer Channel Lineup Container dc:title property should be set to the value of the upnp:channelGroupName property when included.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	3E7UL	N

NOTE: The upnp:channelGroupName property contains the user-friendly name for the Channel Lineup Container. Duplicating this value in the dc:title will provide a user-friendly (and hopefully informative) name to UPnP MediaServer control points that do not understand the upnp:channelGroupName property.

7.4.5.3.10

[GUIDELINE] A UPnP AV MediaServer Channel Lineup Container shall contain CDS Items that are either all streamable (Streamable Channel Object) or all non-streamable (Non-Streamable Channel Object).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	8N7XW	N

NOTE: This requirement clarifies that a Channel Lineup Container cannot contain a mixture of both streamable and non-streamable CDS items. Non-Streamable Channel Objects in a Channel Lineup Container would be used for Extended Tuners that do not stream their content over the network, but provides a CDS object IDs for internal Tuner resources that can be used to setup a Scheduled Recording with identifying channel information (cdsNonEPG record class). Optionally, Non-Streamable Channel Objects in a Channel Lineup Container could be used for Extended Tuners that implement Virtual Tuner Objects in a Virtual Tuner Container 7.4.5.9(7.3.538) to stream selected channel content over the network through a single connection.

7.4.5.3.11

[GUIDELINE] A UPnP AV MediaServer CDS Item (i.e. Non-Streamable Channel Objects or Streamable Channel Objects) contained within a Channel Lineup Container shall conform to the requirements as defined in 7.4.5.3.12 – 7.4.5.3.22.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	BC57O	N

NOTE: This defines the common requirements for CDS Items that represent available channels whether they are non-streamble (Non-Streamable Channel Objects) or streamable (Streamable Channel Objects).

7.4.5.3.12

[GUIDELINE] A UPnP AV MediaServer shall set the upnp:class property value of a Non-Streamable Channel Object or Streamable Channel Object to one of the following:

- object.item.videoItem.videoBroadcast
- object.item.audioItem.audioBroadcast
- class derived from either of the above classes

The upnp:class value shall match the DLNA Media Class of the content carried by either the Non-Streamable Channel Object or the Streamable Channel Object (i.e. supports content streaming) as defined in 7.4.5.5 of these guidelines.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	4I38R	N

NOTE: A UPnP AV MediaServer needs to select a upnp:class value consistent with the defined DLNA Media Classes (Audio Only or Audio Video).

7.4.5.3.13

[GUIDELINE] A UPnP AV MediaServer should expose at least one upnp:channelID property for each Non-Streamable Channel Object or Streamable Channel Object.

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[46]	T3WE7	N

NOTE: A UPnP AV MediaServer Non-Streamable Channel Object or Streamable Channel Object would expose the upnp:channelID property whether the upnp:class property is object.item.videoItem.videoBroadcast, object.item.audioItem.audioBroadcast, or their derived classes.

7.4.5.3.14

[GUIDELINE] A UPnP AV MediaServer Non-Streamable Channel Object or Streamable Channel Object that includes a value for the upnp:channelID property shall include the upnp:channelID@type property.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	NYKYZ	N

7.4.5.3.15

[GUIDELINE] If a UPnP AV MediaServer exposes a upnp:channelID property and a upnp:channelID@type property for a Non-Streamable Channel Object or Streamable Channel Object, then the combination of the upnp:channelID, upnp:channelID@type, and upnp:channelID@distriNetworkID (if exposed) property values shall be unique within all the exposed Channel Lineup Containers.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	U4XE8	N

NOTE: The upnp:channelID property, together with the upnp:channelID@type and optionally the upnp:channelID@distriNetworkID property, uniquely identifies a Non-Streamable Channel Object or Streamable Channel Object. Per [46], when the upnp:channelId@distriNetworkID property is exposed it needs to be exposed for every upnp:channelID property in the CDS. Therefore uniqueness will be determined by either upnp:channelID and upnp:channelID@type property pairs or upnp:channelID, upnp:channelID@type, and upnp:channelID@distriNetworkID property triplets The upnp:channelID property is used to identify a video broadcast channel within the Extended Tuner. It is also used in the CDS Items in the EPG Server Device Option to identify the associated Non-Streamable Channel Object or Streamable Channel Object and in the recordSchedule and recordTask objects in the Scheduled Recording Device Option to identify the associated Non-Streamable Channel Object or Streamable Channel Object.

Note that the upnp:channelID@distriNetworkID property was defined in AVv3 as an additional channelID qualifier.

7.4.5.3.16

[GUIDELINE] If a UPnP AV MediaServer exposes a upnp:channelID property for a Non-Streamable Channel Object or Streamable Channel Object, and the upnp:channelID@type property value is either “ANALOG” or “DIGITAL”, then it should expose the upnp:channelNr property.

[ATTRIBUTES]

S	C	DMS	M-DMS	n/a	[46]	CWJWS	N

NOTE: This maintains compatibility with the DLNA Basic Tuner guideline 7.4.1.4.19.1.

7.4.5.3.17

[GUIDELINE] If a UPnP AV MediaServer exposes a upnp:channelID property for a Non-Streamable Channel Object or Streamable Channel Object, and the upnp:channelID@type property is not “ANALOG” or “DIGITAL”, then it shall not expose the upnp:channelNr property.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	KRMVP	N

NOTE: This is repeating a clarification from [46].

7.4.5.3.18

[GUIDELINE] If a UPnP AV MediaServer exposes the upnp:channelNr property, upnp:channelID property, and a upnp:channelID@type property with the value of “DIGITAL” for a Non-Streamable Channel Object or Streamable Channel Object, then the upnp:channelNr property value shall be set to the same value as the major channel number in the upnp:channelID property.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	5BXZZ	N

NOTE: This is repeating a clarification from [46].

7.4.5.3.19

[GUIDELINE] If a UPnP AV MediaServer exposes the upnp:channelNr property, upnp:channelID property, and a upnp:channelID@type property with the value of “ANALOG” for a Non-Streamable Channel Object or Streamable Channel Object, then the upnp:channelNr property value shall be set to the same value as the upnp:channelID property.

[Attributes]

M	R	DMS	M-DMS	n/a	[46]	Z3YWP	N

NOTE: This is repeating a clarification from [46].

7.4.5.3.20

[Guideline] If a UPnP AV MediaServer exposes the upnp:channelID property with a value of "SI" for the upnp:channelID@type property, then the following definitions shall apply to the value for the upnp:channelID property:

- The <Network ID> term is a non-negative numerical value with a range of 0 to 0xFFFF. The value is network specific and shall be represented as a decimal or hexadecimal value. The value shall be omitted (i.e. empty string) to indicate an unknown <network ID> term.
- The <Transport Stream ID> term is a non-negative numerical value with a range of 0 to 0xFFFF. The value is network specific and shall be represented as a decimal or hexadecimal value. The value shall be omitted (i.e. empty string) to indicate an unknown <Transport Stream ID> term.
- The <Service ID> term is a non-negative numerical value with a range of 0 to 0xFFFF. The value is network specific and can be represented as a decimal or hexadecimal value.

[Attributes]

M	C	DMS	M-DMS	n/a	[46]	Z6TAR	N

NOTE: This provide clarification for the values contained within the upnp:channelID property for the "SI" type that's lacking in [46]. Examples of valid values for the upnp:channelID are as follows:

- `<upnp:channelID type="SI">0x1234,0xFEDC,0x0102</upnp:channelID>`
- `<upnp:channelID type="SI">12345,23456,32109</upnp:channelID>`
- `<upnp:channelID type="SI">,1,0x0102</upnp:channelID>`
- `<upnp:channelID type="SI>,,0x0102</upnp:channelID>`

7.4.5.3.21

[Guideline] If a UPnP AV MediaServer exposes the upnp:channelID property with the upnp:channelID@type property with a value of "DLNA.ORG_FPF", then the value of the upnp:channelID property shall contain a CSV (Comma Separated Value List) triplet containing values for frequency, program number, and modulation format respectively, where the frequency value contains the channel frequency in Hertz, the program number value is a 16-bit value as defined in [5] and shall be represented as a decimal or hexadecimal value, and the modulation format value contains a vendor defined string representing the modulation format being used. Valid values for the modulation format are defined in the following table:

Table 41 — Modulation Format Values

Broadcast Systems	Modulation Format Values
US Terrestrial System	ATSC-8VSB
US Cable System	SCTE65-QPSK SCTE65-BPSK SCTE65-OQPSK SCTE65-VSB8 SCTE65-VSB16 SCTE65-QAM16 SCTE65-QAM32 SCTE65-QA64M SCTE65-QAM80 SCTE65-QAM96 SCTE65-QAM112 SCTE65-QAM128 SCTE65-QAM160

	SCTE65-QAM192 SCTE65-QAM224 SCTE65-QAM256 SCTE65-QAM320 SCTE65-QAM384 SCTE65-QAM448 SCTE65-QAM512 SCTE65-QAM640 SCTE65-QAM768 SCTE65-QAM896 SCTE65-QAM1024

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[5] [46]	WNKRA	N

NOTE: This is a DLNA extension to the upnp:channelID@type property values as defined in appendix B.8.5 of [46].

Examples for upnp:channelID values are as follows:

- `<upnp:channelID type="DLNA.ORG_FPF">867000000,0x0002, SCTE65-QAM32</upnp:channelID>`
- `<upnp:channelID type="DLNA.ORG_FPF">867000000,20,ATSC-8VSB</upnp:channelID>`

7.4.5.3.22

[GUIDELINE] A UPnP AV MediaServer should implement Basic Tuner guideline requirement 7.4.1.4.15.6 which returns Non-Streamable Channel Objects or Streamable Channel Objects in the order that corresponds to the up / down operation.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	YH5P3	N

NOTE: The order of the <item> elements in a CDS:Browse response (without a Sort input specified) should represent order of an up / down "channel surfing" action to allow UPnP AV MediaServer control points to provide this functionality.

7.4.5.4 MM/ET Non-Streamable Extended Tuner Guidelines

7.4.5.4.1

[GENERAL] These guidelines define the baseline requirements for Non-Streamable Channel Objects. Non-Streamable Channel Objects are non-streamable CDS objects (i.e. no res property value) which represent a single channel of a broadcast source which presents content in a "channelized" format. Implementers should note that Non-Streamable Channel Objects are not restricted to representing traditional terrestrial, cable, or satellite broadcast channels and can be used to represent Webcasts, so called "Internet Radio" and "Internet TV" stations, and other emerging content delivery mediums, so long as the content is organized or presented to the user through the upnp:channelID property.

7.4.5.4.2

[GUIDELINE] If a UPnP AV MediaServer exposes Non-Streamable Channel Objects (i.e. non-streamable), then it shall conform to guideline requirements 7.4.5.4.3 – 7.4.5.4.4.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	JUT5C	N

7.4.5.4.3

[GUIDELINE] A UPnP AV MediaServer that advertises support for the TUNER Feature as defined in 7.4.5.3.2 through 7.4.5.3.4 shall not include the dlna:containerType property (7.4.1.4.15.4) in any Channel Lineup Containers that contain only Non-Streamable Channel Objects.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	N4X7P	N

NOTE: UPnP AV MediaServer control points can differentiate between implementation levels by looking for the presence or absence of the dlna:containerType property.

7.4.5.4.4

[GUIDELINE] A UPnP AV MediaServer Non-Streamable Channel Object shall omit the URI value from all the res properties.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	A9D7O	N

NOTE: Since these CDS items are not streamable, a res property is not needed. But having res properties without a URI value (a non streamable Channel Object can have multiple res elements) can be useful by UPnP AV MediaServer control points in determining the DLNA media format profile for the Non-Streamable Channel Object by examining the res@protocolInfo property's DLNA.ORG_PN value in the fourth field

7.4.5.5 MM/ET Streamable Extended Tuner Guidelines

7.4.5.5.1

[GUIDELINE] These guidelines define the baseline requirements for Streamable Channel Objects. Streamable Channel Objects are streamable CDS objects which represent a single channel of a broadcast source which presents content in a "channelized" format. Implementers should note that Streamable Channel Objects are not restricted to representing traditional terrestrial, cable, or satellite broadcast channels and can be used to represent Webcasts, so called "Internet Radio" and "Internet TV" stations, and other emerging content delivery mediums, so long as the content is organized or presented to the user through the upnp:channelID property. Streamable Channel Objects expose one or more res properties with URI values. Content Receivers establish a connection to the URI to initiate a streaming session. This connection implicitly attempts to change the channel of the underlying broadcast source to the channel represented by the Streamable Channel Object. Whether the channel is changed and the streaming connection established is ultimately determined by the vendor-specific arbitration logic which is outside the scope of these guidelines. UPnP AV MediaServer control points can switch the channel of the broadcast source by establishing a connection to the URI in a different Streamable Channel Object.

7.4.5.5.2

[GUIDELINE] If a UPnP AV MediaServer exposes Streamable Channel Objects, then it shall conform to guideline requirements 7.4.5.5.3 – 7.4.5.5.8.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	FIDAZ	N

7.4.5.5.3

[GUIDELINE] A UPnP AV MediaServer Streamable Channel Object shall expose one or more res properties. Each res property shall include a valid URI value.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	4DHUP	N

NOTE: A Streamable Channel Object is identified as a CDS Item with one or more res properties with a valid URI value

7.4.5.5.4

[GUIDELINE] A UPnP AV MediaServer that advertises support for the TUNER Feature as defined in 7.4.5.3.2 through 7.4.5.3.4 shall include the dlna:containerType property and conform to guideline 7.4.1.4.15.4 in any Channel Lineup Container that contains only Streamable Channel Objects.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	VJFVP	N

NOTE: UPnP AV MediaServer control points can differentiate between implementation levels by looking for the presence or absence of the dlna:containerType property. This is to ensure backwards compatibility with UPnP AV MediaServer control points that will only interoperate with Basic Tuners.

7.4.5.5.5

[GUIDELINE] A UPnP AV MediaServer implementation for a Streamable Extended Tuner shall conform to Basic Tuner guidelines defined in 7.4.1.4.15 to 7.4.1.4.21.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	M5B39	N

NOTE: This is to ensure backwards compatibility with UPnP AV MediaServer control points that will only interoperate with Basic Tuners. This implies that a Streamable Extended Tuner is a superset of a Basic Tuner

7.4.5.5.6

[GUIDELINE] A UPnP AV MediaServer shall support at least one streaming connection for a Streamable Channel Object. A UPnP AV MediaServer can refuse a connection if the underlying broadcast source is unavailable due to resource limitations associated with other active streaming connections or local playback operations.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46]	UTQBZ	N

NOTE: A UPnP AV MediaServer will not have sufficient resources sometimes to support multiple streaming connections to a single Streamable Channel Object. If a UPnP AV MediaServer cannot accept a new streaming connection it can refuse the connection as defined in the relevant Media Transport guideline (e.g. 7.5.4.3.2.4.2).

7.4.5.5.7

[GUIDELINE] A UPnP AV MediaServer shall terminate all streaming connections if the channel of the underlying broadcast source changes and is no longer represented by the Streamable Channel Object.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46]	N6C39	N

NOTE: A UPnP AV MediaServer control point establishes a connection to a Streamable Channel Object for purposes of steaming that channel. If the UPnP AV MediaServer is no longer able to stream that channel's content, it needs to terminate any active streaming connections.

7.4.5.5.8

[GUIDELINE] If a UPnP AV MediaServer implements a "time shift buffer" or similar feature on the broadcast source associated with a Streamable Channel Object, then this buffer should be exposed using the DLNA UCDAM model.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	JRP9S	N

NOTE: The DLNA UCDAM buffer model is designed to allow a UPnP AV MediaServer control point to interoperate with “time shift buffers” and similar features. Vendors should pay particular attention to guidelines in 7.4.1.3.32 regarding the UCDAM s_0 boundary increasing, and guideline in 7.4.1.3.33 describing the UCDAM s_n boundary increasing requirements.

7.4.5.6 MM/ET Presets Containers

7.4.5.6.1

[General] These guidelines define the optional Presets Containers. Many broadcast receivers utilize objects which can be described as “Favorites”, “Presets”, or other terms, which associate a user-defined set of channel identifiers with channel objects. Vendors can utilize one or more Presets Containers to communicate these mappings or associations to a UPnP AV MediaServer control point. A Presets Container needs to have ContentDirectory service Reference Items which associate channel identifiers with channel objects.

7.4.5.6.2

[Guideline] A UPnP AV MediaServer may expose a Presets Container.

[Attributes]

O	A	DMS	M-DMS	n/a	[46]	DSZKQ	N

7.4.5.6.3

[Guideline] If a UPnP AV MediaServer exposes a Presets Container, then it shall conform to the guidelines in 7.4.5.6.4 – 7.4.5.6.10.

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	PYVZ5	N

7.4.5.6.4

[Guideline] A UPnP AV MediaServer shall set the upnp:class property value of a Presets Container to one of the following:

- object.container.channelGroup
- object.container.channelGroup.audioChannelGroup
- object.container.channelGroup.videoChannelGroup
- class derived from any of the above classes

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	7X69R	N

NOTE: [46] Requires channel containers to have the upnp:class of object.container.channelGroup or any of its derived classes, which gives more precise guidance on upnp:class usage than the requirements in the guideline 7.4.1.4.15.3 for a Basic Tuner.

[46] Appendix C.2.2.5 specifies that object.container.channelGroup.audioChannelGroup only contains object.item.audioItem.audioBroadcast items and object.container.channelGroup.videoChannelGroup only contains object.item.videoItem.videoBroadcast items.

7.4.5.6.5

[Guideline] A UPnP AV MediaServer Presets Container that contains @refID properties to CDS Items (7.4.5.6.10) that are both audioBroadcast and videoBroadcast items shall set the upnp:class property value to object.container.channelGroup or a derived class of object.container.channelGroup.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[46]	7ZYLU	N

NOTE: [46] Profiles this guidance.

7.4.5.6.6

[GUIDELINE] A UPnP AV MediaServer Presets Container shall include a value for the upnp:channelGroupName property.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	5IQ4C	N

NOTE: A channel group defines a group of channels. A device that has multiple tuners may provide multiple channel groups. Moreover, a physical tuner device may provide multiple channel groups (for example, a set-top-box that contains a single tuner but supports three different input connections: terrestrial, cable, and satellite). Note that this mandates a similar requirement for Channel Lineup Container. The upnp:channelGroupName property value represents the user-friendly name of the Presets Container as decribed in Appendix B.9.1 of [46].

7.4.5.6.7

[GUIDELINE] A UPnP AV MediaServer Presets Container shall include the upnp:channelGroupName@id property which shall be set to the value DLNA.ORG_ Presets for Presets Containers.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	7CB26	N

NOTE: The value “DLNA.ORG_ Presets” identifies the channelGroup container as a Presets Container to UPnP AV MediaServer control points.

7.4.5.6.8

[GUIDELINE] A UPnP AV MediaServer Presets Container dc:title property should be set to the value of the upnp:channelGroupName property.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	6JR4D	N

NOTE: The upnp:channelGroupName property contains the user-friendly name for the Presets Container. Duplicating this value in the dc:title will provide a user-friendly (and hopefully informative) name to UPnP MediaServer control points that do not understand the upnp:channelGroupName property.

7.4.5.6.9

[GUIDELINE] A UPnP AV MediaServer Presets Container shall not include the dlna:containerType property (7.4.1.4.15.4).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	UA9XW	N

7.4.5.6.10

[GUIDELINE] A UPnP AV MediaServer Presets Container shall expose a @refID property in each CDS Item contained in a Presets Container. The @refID property shall contain a value equal to the @id property value of a Non-Streamable Channel Object or Streamable Channel Object exposed in the same ContentDirectory service.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	V8GLU	N

NOTE: The purpose of a Preset reference is to associate a user-friendly “channel identifier” with a Non-Streamable Channel Object or Streamable Channel Object. The UPnP AV MediaServer exposes this user-friendly identifier in the upnp:channelName of a videoBroadcast or audioBroadcast item and associates this with the Non-Streamable Channel Object or Streamable Channel Object using the @refID property.

7.4.5.7 MM/ET EPG Server Device Option Additional Tuner Guidelines

7.4.5.7.1

[GENERAL] These guidelines are additional requirements on an Extended Tuner when a UPnP AV MediaServer implements the EPG Server Device Option.

7.4.5.7.2

[GUIDELINE] If a UPnP AV MediaServer implements the EPG Server Device Option, as defined in 7.4.6.2.3, by including a <Feature> element with the Feature@name attribute equal to “EPG” in the FeatureList output argument in response to the CDS:GetFeatureList action ., then it shall conform to guideline requirements 7.4.5.7.3 – 7.4.5.7.4.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	MNO39	N

7.4.5.7.3

[GUIDELINE] A UPnP AV MediaServer Channel Lineup Container shall include a value for the upnp:channelGroupName property. The value represents the user-friendly name of the Channel Lineup Container.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	PCLAV	N

NOTE: For the EPG Server Device Option, Channel Lineup Containers have to expose the upnp:channelGroupName property.

7.4.5.7.4

[GUIDELINE] A UPnP AV MediaServer shall expose at least one upnp:channelID property for each Non- Streamable Channel Object or Streamable Channel Object.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	4OQNX	N

NOTE: For the EPG Server Device Option, Non- Streamable Channel Objects and Streamable Channel Objects have to expose the upnp:channelID property.

7.4.5.8 MM/ET Scheduled Recording Device Option Additional Tuner Guidelines

7.4.5.8.1

[GENERAL] These guidelines are additional requirements on an Extended Tuner when a UPnP AV MediaServer implements the Scheduled Recording Device Option.

7.4.5.8.2

[GUIDELINE] If a UPnP AV MediaServer implements the Scheduled Recording Device Option, then it shall conform to guideline 7.4.5.8.3.

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	9EJWK	N

7.4.5.8.3

[Guideline] A UPnP AV MediaServer shall implement guideline 7.4.5.3.13 (i.e. upnp:channelID property) as mandatory instead of recommended.

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	I4Z7F	N

NOTE: For the Scheduled Recording Device Option, Non- Streamable Channel Objects and Streamable Channel Objects have to expose the upnp:channelID property.

7.4.5.9 MM/ET Virtual Tuners

7.4.5.9.1

[General] These optional guidelines for an Extended Tuner implementing a Virtual Tuner feature allow modeling a physical tuner through a single connection. A Virtual Tuner Object is capable of streaming the output of multiple channels over a single URI (res property URI) connection.

7.4.5.9.2

[Guideline] A UPnP AV MediaServer may implement a Virtual Tuner as defined in guidelines requirements 7.4.5.9.3 – 7.4.5.9.35.

[Attributes]

O	A	DMS	M-DMS	n/a	[46]	AFKWK	N

7.4.5.9.3

[Guideline] If a UPnP AV MediaServer implements a Virtual Tuner, then it shall conform to the guidelines requirements 7.4.5.9.4 – 7.4.5.9.35.

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	KUQR9	N

7.4.5.9.4

[Guideline] A UPnP AV MediaServer shall set the upnp:class property value of a Virtual Tuner Container to one of the following:

- object.container.virtualStreamGroup
- class derived from the above class

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	Z4AG6	N

7.4.5.9.5

[Guideline] A UPnP AV MediaServer Virtual Tuner Container shall include a value for the upnp:channelGroupName property. The value represents the user-friendly name of the Virtual Tuner Container.

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	DA6UG	N

7.4.5.9.6

[GUIDELINE] A UPnP AV MediaServer Virtual Tuner Container shall include the upnp:channelGroupName@id property which shall be set to the value DLNA.ORG_VirtualTuner for Virtual Tuner Containers.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	927R6	N

NOTE: The value “DLNA.ORG_VirtualTuner” identifies the UPnP container as a Virtual Tuner Container to UPnP AV MediaServer control points.

7.4.5.9.7

[GUIDELINE] A UPnP AV MediaServer Virtual Tuner Container dc:title property should be set to the value of the upnp:channelGroupName property.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	TB9HA	N

NOTE: The upnp:channelGroupName property contains the user-friendly name for the Virtual Tuner Container. Duplicating this value in the dc:title will provide a user-friendly (and hopefully informative) name to UPnP AV MediaServer control points that do not understand the upnp:channelGroupName property.

7.4.5.9.8

[GUIDELINE] A UPnP AV MediaServer Virtual Tuner Container shall not include the dlna:containerType property (7.4.1.4.15.4).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	6LRLP	N

NOTE: This allows UPnP AV MediaServer control points to differentiate that this is not a Basic Tuner.

7.4.5.9.9

[GUIDELINE] CDS items (Virtual Tuner Objects) as defined in guideline 7.4.5.9.10 contained in a UPnP AV MediaServer Virtual Tuner Container shall be streamable through a single connection and shall reflect in the res@protocolInfo 4th field value for the Virtual Tuner Object the content currently being streamed over the connection as defined in 7.4.1.3.15.4 and 7.4.1.3.16.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	SLJVJ	N

NOTE: This requirement clarifies that a Virtual Tuner Container contains CDS items that stream tuner channels through a single connection.

7.4.5.9.10

[GUIDELINE] A UPnP AV MediaServer CDS Item (i.e. Virtual Tuner Object) contained within a Virtual Tuner Container shall conform to the requirements as defined in 7.4.5.9.11 – 7.4.5.9.35.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	RSZXO	N

NOTE: This defines the requirements for a CDS Item that represents the current streamed output of a tuner (Virtual Tuner Objects).

7.4.5.9.11

[GUIDELINE] A UPnP AV MediaServer shall set the upnp:class property value of a Virtual Tuner Object to one of the following:

- object.item.videoItem.virtualTuner
- object.item.audioItem.virtualTuner
- class derived from either of the above classes

The upnp:class value shall match the DLNA Media Class of the content carried by the Virtual Tuner Object.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	VDBHA	N

NOTE: A UPnP AV MediaServer needs to select a upnp:class value consistent with the defined DLNA Media Classes (Audio Only or Audio Video).

7.4.5.9.12

[GUIDELINE] A UPnP AV MediaServer shall expose the dlna:channelGroupList property for each Virtual Tuner Object. The value of the dlna:channelGroupList property is a CSV (Comma Separated Value) list containing one or more upnp:channelGroup@id values, where each upnp:channelGroup@id value corresponds to a Channel Lineup Container that the Virtual Tuner Object can use for access to a Tuner.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	XU9MQ	N

NOTE: This guideline provides UPnP AV MediaServer control points a mechanism to know which Channel Lineup Container(s) are associated with which Virtual Tuner Object.

7.4.5.9.13

[GUIDELINE] A UPnP AV MediaServer shall expose at least one upnp:channelID property for each Virtual Tuner Object. The upnp:channelID property value represents the channel last selected by the Virtual Tuner Object. When more than one upnp:channelID property is exposed, they shall each resolve to the same underlying channel, but represented differently (i.e. different upnp:channelID@type property values as defined in 7.4.5.9.14).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	BTGLV	N

NOTE: The values for multiple instances of the upnp:channelID property represent the same underlying channel currently being selected.

7.4.5.9.14

[GUIDELINE] A UPnP AV MediaServer Virtual Tuner Object that includes a value for the upnp:channelID property shall include the upnp:channelID@type property.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	VRLN5	N

7.4.5.9.15

[GUIDELINE] If a UPnP AV MediaServer exposes more than one upnp:channelID property for a Virtual Tuner Object then each upnp:channelID property shall have a distinct upnp:channelID@type property value.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	KXMVM	N

NOTE: Each upnp:channelID property exposed in a Virtual Tuner Object will always have different upnp:channelID@type values.

7.4.5.9.16

[GUIDELINE] If a UPnP AV MediaServer's upnp:channelID@type property value is either "ANALOG" or "DIGITAL" for a Virtual Tuner Object, then it should expose the upnp:channelNr property.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	Y49CI	N

NOTE: This is aligning with a clarification from[46].

7.4.5.9.17

[GUIDELINE] If a UPnP AV MediaServer's upnp:channelID@type property is not "ANALOG" or "DIGITAL" for a Virtual Tuner Object, then it shall not expose the upnp:channelNr property.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	OYZ6Q	N

NOTE: This is aligning with a clarification from[46].

7.4.5.9.18

[GUIDELINE] If a UPnP AV MediaServer exposes the upnp:channelNr property, upnp:channelID property, and a upnp:channelID@type property with the value of "DIGITAL" for a Virtual Tuner Object, then the upnp:channelNr property value shall be set to the major channel number from that upnp:channelID property

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	YKSPG	N

NOTE: This is aligning with a clarification from[46].

7.4.5.9.19

[GUIDELINE] If a UPnP AV MediaServer exposes the upnp:channelNr property, upnp:channelID property, and a upnp:channelID@type property with the value of "ANALOG" for a Virtual Tuner Object, then the upnp:channelNr property value shall be set to the value of that upnp:channelID property.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	723RJ	N

NOTE: This is aligning with a clarification from[46].

7.4.5.9.20

[GUIDELINE] A UPnP AV MediaServer should implement the DLNA defined value of "DLNA.ORG_OBJECT_ID" for the upnp:channelID@type property as defined in 7.4.5.9.21.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	QUV45	N

7.4.5.9.21

[GUIDELINE] If a UPnP AV MediaServer Virtual Tuner Object exposes the upnp:channelID property with the upnp:channelID@type property with a value of "DLNA.ORG_OBJECT_ID", then the value of the upnp:channelID property shall contain the @id property value of a Streamable Channel Object as defined in guideline requirements 7.4.5.3.11 and 7.4.5.5. .

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	L943V	N

NOTE: This is a DLNA extension to the upnp:channelID@type property values as defined in appendix B.8.5 of [46].

7.4.5.9.22

[GUIDELINE] A UPnP AV MediaServer Virtual Tuner Object shall expose the following physical tuner status-related properties:

- upnp:tuned
- upnp:signalStrength
- upnp:signalLocked

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	YACS6	N

NOTE: These UPnP properties are mandated for all Virtual Tuner Objects.

7.4.5.9.23

[GUIDELINE] A UPnP AV MediaServer that exposes a Virtual Tuner Object shall implement the CDS:X_DLNA_SelectChannel action with the following syntax:

The action is defined as follows:

Table 42 — CDS:X_DLNA_SelectChange Action Parameters

Argument	Direction	relatedStateVariable
VirtualTunerObjectID	In	A_ARG_TYPE_ObjectID
ChannelID	In	A_ARG_TYPE_DLNAChannelID
ClientConnectionID	In	A_ARG_TYPE_DLNAConnectionID
ServerConnectionID	Out	A_ARG_TYPE_DLNAConnectionID

The value of the VirtualTunerObjectID input argument shall be the object ID of a Virtual Tuner Object.

The value of the ChannelID input argument shall be a upnp:channelID XML Fragment. See Appendix B.8.5 of [46] for details with the exception that only a single instance is allowed (i.e. not multi-valued). The ChannelID input argument shall be in escaped XML when used with this action.

The value of the ClientConnectionID input argument shall be a value obtained by any of the following methods:

- The ServerConnecrtionID output argument value from a previous CDS:X_DLNA_SelectChannel action response.
- The ConnectionID output argument value from a CMS:PrepareForConnection action response.
- -1 when this action is used to establish a new connection.

The value of the ServerConnectionID output argument shall be the same value as the ClientConnectionID unless the ClientConnectionID input argument value is -1, in which case the UPnP AV MediaServer shall respond with a ServerConnectionID value that is unique to the connection.

When the ClientConnectionID input argument is -1 the UPnP AV MediaServer control point is initiating a new connection and requesting a unique connection ID in the ServerConnectionID output argument in the action response. If the action succeeds the UPnP AV MediaServer shall use the value returned in the ServerConnectionID output argument for BCM operations.

The error codes defined for this action are defined in Table 43:

Table 43 — CDS:X_DLNA_SelectChange Action Error Codes

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control [].
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
701	No such object	The X_DLNA_SelectChannel action failed because the specified VirtualTunerObjectID does not exist or is not a Virtual Tuner Object.
712	Bad metadata	The X_DLNA_SelectChannel action failed because the ChannelID input argument is an invalid XML Fragment.
715	Source resource access denied	The X_DLNA_SelectChannel action failed because a specified ChannelID resource is busy.
720	Cannot process the request	The X_DLNA_SelectChannel action failed because it will not be able to tune to the channel specified in the ChannelID resource.
801	Invalid connection reference	The X_DLNA_SelectChannel action failed because the connection reference argument does not refer to a valid connection established by the Connection Manager Service (CMS).

This action shall be defined in the service description document using the following XML fragment:

```
<action>
   <name> X_DLNA_SelectChannel </name>
   <argumentList>
      <argument>
         <name>VirtualTunerObjectID</name>
         <direction>in</direction>
         <relatedStateVariable>A_ARG_TYPE_ObjectID</relatedStateVariable>
```

```
        </argument>
        <argument>
            <name>ChannelID</name>
            <direction>in</direction>
            <relatedStateVariable>A_ARG_TYPE_DLNAChannelID</relatedStateVariable>
        </argument>
        <argument>
            <name>ClientConnectionID</name>
            <direction>in</direction>
            <relatedStateVariable>A_ARG_TYPE_DLNAConnectionID</relatedStateVariable>
        </argument>
        <argument>
            <name>ServerConnectionID</name>
            <direction>out</direction>
            <relatedStateVariable>A_ARG_TYPE_DLNAConnectionID</relatedStateVariable>
        </argument>
    </argumentList>
</action>
```

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	UDADM	N

NOTE: This guideline defines a new action for the UPnP AV MediaServer ContentDirectory service. This action is implemented by servers that expose Virtual Tuner Objects. Semantic behavior for this action is defined in 7.4.5.9.24 – 7.4.5.9.32.

Examples of valid values for the ChannelID input argument are as follows:

- `<upnp:channelID xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/" type=”ANALOG”>5</upnp:channelID>`
- `<upnp:channelID xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/" type=”DLNA.ORG_OBJECT_ID”>TunerObj_001</upnp:channelID>`

The values of the ClientConnectionID input argument and the ServerConnectionID output argument allow the CDS:X_DLNA_SelectChannel action to be used as a new connection initiator or to be associated with an existing connection.

7.4.5.9.24

[GUIDELINE] A UPnP AV MediaServer control point should first verify if the Virtual Tuner Object is currently in use in an active session with another UPnP AV MediaRenderer (e.g. dlna:peerManager property as defined in 7.4.5.9.35) before executing the X_DLNA_SelectChannel action.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	2KYHG	N

NOTE: This guideline provides implementation guidance for UPnP AV MediaServer control points to be a good citizen on the network.

7.4.5.9.25

[GUIDELINE] A UPnP AV MediaServer that exposes a Virtual Tuner Object shall include the A_ARG_TYPE_DLNAChannelID state variable in the ContentDirectory service description. This state variable shall be defined as type information for the ChannelID input argument in the CDS:X_DLNA_SelectChannel action.

This state variable is defined in Table 44:

Table 44 — A_ARG_TYPE_DLNAChannelID State Variable

Variable Name	Data Type	Allowed Value	Evented	Moderated Event
A_ARG_TYPE_DLNAChan	string	upnp:channelID	No	No

nelID		XML Fragment See Appendix B.8.5 of [46] with the exception that only a single instance is allowed		

The A_ARG_TYPE_DLNAChannelID state variable shall be defined in the service description document using the following XML fragment:

```
<stateVariable sendEvents=”no”>
   <name>A_ARG_TYPE_DLNAChannelID</name>
   <dataType>string</dataType>
</stateVariable>
```

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	YEMMQ	N

NOTE: This guideline defines a new state variable for the UPnP AV MediaServer ContentDirectory service. This state variable is used by the new CDS:X_DLNA_SelectChannel action for the ChannelID input argument. Example values for this state variable are as follows:

- `<upnp:channelID xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/" type=”ANALOG”>5</upnp:channelID>`
- `<upnp:channelID xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/" type=”DLNA.ORG_OBJECT_ID”>TunerObj_001</upnp:channelID>`

7.4.5.9.26

[GUIDELINE] A UPnP AV MediaServer that exposes a Virtual Tuner Object shall include the A_ARG_TYPE_DLNAConnectionID state variable in the ContentDirectory service description. This state variable shall be defined as type information for the ClientConnectionID input argument and the ServerConnectionID output argument in the CDS:X_DLNA_SelectChannel action.

This state variable is defined in Table 45:

Table 45 — A_ARG_TYPE_DLNAConnectionID State Variable

Variable Name	Data Type	Allowed Value	Evented	Moderated Event
A_ARG_TYPE_DLNAConnectionID	i4	Connection ID value established by a UPnP AV MediaServer. The value -1 is a special value indicating an unsolicited action requesting a new connection.	No	No

The A_ARG_TYPE_DLNAConnectionID state variable shall be defined in the service description document using the following XML fragment:

```
<stateVariable sendEvents=”no”>
   <name>A_ARG_TYPE_DLNAConnectionID</name>
   <dataType>i4</dataType>
</stateVariable>
```

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	GVCIJ	N

NOTE: This guideline defines a new state variable for the UPnP AV MediaServer ContentDirectory service. This state variable is used by the new CDS:X_DLNA_SelectChannel action for the ClientConnectionID input argument and ServerConnectionID output argument.

7.4.5.9.27

[GUIDELINE] If a UPnP AV MediaServer has successfully executed the CDS:X_DLNA_SelectChannel action and the content is being streamed or available to be streamed, then it shall expose the upnp:tuned property with a value of "1" for the Virtual Tuner Object, otherwise the value shall be "0"

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	GFP9J	N

NOTE: This provides UPnP AV MediaServer control points a method to determine if content (e.g. channel) is currently being streamed or available to be streamed by the Virtual Tuner Object.

7.4.5.9.28

[GUIDELINE] If a UPnP AV MediaServer has successfully executed the CDS:X_DLNA_SelectChannel action, then it should only return a success response after it has stopped streaming content for the previously selected channel.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	TNROB	N

NOTE: This allows UPnP AV MediaServer control points to know that content from the previously selected channel has completed streaming content to a rendering device.

Note that if a DLNA media format profile supports DIT (Discontinuity Information Table), then the DIT is inserted into the content stream at the discontinuity point as defined in Media Format Guidelines[56]. This provides a mechanism for a UPnP AV MediaRenderer to detect a change in channel content.

7.4.5.9.29

[GUIDELINE] If a UPnP AV MediaServer control point has successfully executed the CDS:X_DLNA_SelectChannel action, then its co-located UPnP AV MediaRenderer control point should execute the AVT:SetAVTransportURI action with the CurrentURIMetadata input parameter containing the updated CDS metadata of the Virtual Tuner Object resulting from the CDS:X_DLNA_SelectChannel action to the UPnP AV MediaRenderer that is rendering the content being streamed as defined in 7.4.5.9.27.

[ATTRIBUTES]

S	A	DMC	M-DMC	n/a	[46]	UWUZ3	N

NOTE: In the 3-box System Usage, the UPnP AV MediaRenderer needs to have knowledge of the updated metadata (i.e. channelID value) when the CDS:X_DLNA_SelectChannel action is successfully executed. The updated CDS metadata is obtained by executing a CDS:Browse action on the Virtual Turner Object after executing the CDS:X_DLNA_SelectChannel action.

A possible implementation of a 3-box System Usages scenario to achieve a channel change user experience that is consistent with the commercial TV services offered by content service providers is as follows: Upon a user's request for a channel change, the UPnP AV control point (i.e. UPnP AV MediaRenderer + UPnP AV MediaServer control point) invokes the AVT:Stop action to a UPnP AV MediaRenderer (e.g. DMR), followed by the CDS:X_SelectChannel action to a UPnP AV MediaServer (e.g. DMS) to tune to a new channel. After the UPnP AV MediaServer returns a successful response, the UPnP AV control point invokes the CDS:Browse action on the Virtual Turner Object to obtain the updated CDS metadata, followed by invoking the AVT:SetAVTransportURI action to the UPnP AV MediaRender with the updated content item metadata, and then finally invokes the AVT:Play action to the UPnP AV MediaRenderer.

7.4.5.9.30

[GUIDELINE] If a UPnP AV MediaServer has successfully executed the CDS:X_DLNA_SelectChannel action, then it shall remove all instances of the upnp:channelID property for the Virtual Tuner Object as specified in the VirtualTunerObjectID input parameter and it shall expose a upnp:channelID property with the value in the ChannelID input parameter.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	GV22A	N

NOTE: This exposes the new channel that has been selected in the Virtual Tuner Object.

7.4.5.9.31

[GUIDELINE] If a UPnP AV MediaServer has successfully executed the CDS:X_DLNA_SelectChannel action, then it may expose additional upnp:channelID property values that conforms to guideline 7.4.5.9.13.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46]	633AS	N

NOTE: This allows a UPnP AV MediaServer to expose additional instances of the upnp:channelID property when it resolves to the same underlying channel as the ChannelID input parameter.

7.4.5.9.32

[GUIDELINE] If a UPnP AV MediaServer has successfully executed the CDS:X_DLNA_SelectChannel action, then it shall update the res@protocolInfo 4th field value for the Virtual Tuner Object as specified in the VirtualTunerObjectID input parameter to identify the content for the selected channel as defined in 7.4.1.3.15.4 and 7.4.1.3.16.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	J7NPO	N

NOTE: This requires the res@protocolInfo property to reflect the correct Media Format Profile for the content on the newly selected channel as specified in the ChannelID input parameter.

7.4.5.9.33

[GUIDELINE] A UPnP AV MediaServer that exposes a Virtual Tuner Object shall implement the BCM (Basic Connection Management) feature as defined in guidelines defined in 7.4.1.5.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	A4FJW	N

NOTE: This guideline requires a UPnP AV MediaServer to implement the BCM feature when implementing Virtual Tuners.

7.4.5.9.34

[GUIDELINE] A UPnP AV MediaServer that exposes the dlna:peerManager property shall have the following definition:

- Namespace: dlna
- Property data type: xsd:string
- Multi-Value: NO
- The dlna:peerManager property semantic behavior is defined and exposed as defined in 7.4.5.9.35.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	JO94M	N

NOTE: This describe a new DLNA namespace property for CDS items.

7.4.5.9.35

[Guideline] If a UPnP AV MediaServer exposes a Virtual Tuner Object that is being used for content transfers, then it shall expose the dlna:peerManager property with one of the follow values:

- The PeerConnectionManager value provided by a UPnP AV MediaRenderer (i.e. DMR) using the peerManager.dlna.org HTTP header (defined in 7.5.4.3.2.36) when the Virtual Tuner Object is currently being used for content transfer.
- The PeerConnectionManager value provided by a UPnP AV MediaRenderer (i.e. DMR) using the peerManager.dlna.org RTSP header (defined in 7.5.4.4.6.2.27) when the Virtual Tuner Object is currently being used for content transfer.
- An empty value if the information is not available (i.e. the UPnP AV MediaRenderer or UPnP AV MediaRenderer control point does not implement BCM).
- The friendly name value provided by a UPnP AV MediaServer control point with embedded rendering (i.e. DMP and M-DMP) using the friendlyName.dlna.org HTTP header (defined in 7.5.4.3.2.37) when the Virtual Tuner Object is currently being used for content transfer.
- The friendly name value provided by a UPnP AV MediaServer control point with embedded rendering (i.e. DMP and M-DMP) using the friendlyName.dlna.org RTSP header (defined in 7.5.4.4.6.2.28) when the Virtual Tuner Object is currently being used for content transfer.

Otherwise, the UPnP AV MediaServer shall not expose the dlna:peerManager property for the Virtual Tuner Object.

[Attributes]

M	A	DMS	M-DMS	n/a	[46]	CBPIN	N

NOTE: This allows a UPnP AV MediaRenderer control point to know which device is currently rendering the content being transferred by the Virtual Tuner Object.

7.4.6 EPG Media Management Guidelines

7.4.6.1 MM/EPG Foreign Metadata Feature Advertisement

7.4.6.1.1

[Guideline] If one or more CDS items exposes foreign metadata using the upnp:ForeignMetadata property, a UPnP AV MediaServer shall include a <Feature> element with the @name attribute equal to "FOREIGN_METADATA" in the FeatureList output argument in response to the CDS:GetFeatureList action. The @version attribute of the same <Feature> element shall have a value of "1".

[Attributes]

M	R	DMS	M-DMS	n/a	[46]	EDCBT	N

7.4.6.2 MM/EPG Server Device Option Advertisement

7.4.6.2.1

[General] This defines the DLNA Electronic Program Guide System Usage guidelines for a UPnP AV MediaServer that implements the EPG Server Device Option

7.4.6.2.2

[GUIDELINE] A UPnP AV MediaServer that implements the EPG Server Device Option shall implement all mandatory requirements of the UPnP EPG Feature as defined in Appendix E.1 of [46].

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	NU8GR	N

7.4.6.2.3

[GUIDELINE] A UPnP AV MediaServer shall include a <Feature> element with the @name attribute equal to “EPG” or “DLNA.ORG_EPGDataOnly” in the FeatureList output argument in response to the CDS:GetFeatureList action. If the EPG Server Device Option is being implemented solely for purpose of providing EPG metadata to other devices, then the Feature@name attribute shall be “DLNA.ORG_EPGDataOnly”. Otherwise, the Feature@name attribute shall be “EPG”. In both cases the @version attribute of the same <Feature> element shall have a value of “1”. The <Feature> element shall include an <objectIDs> element.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	3JAIN	N

NOTE: DLNA.ORG_EPGDataOnly usage is encouraged in cases where a DLNA EPG Server Device Option is used solely as a mechanism for supplying ancillary metadata to control points on the home-network.

7.4.6.2.4

[GUIDELINE] If a UPnP AV MediaServer exposes one or more CDS containers with a upnp:class value of object.container.epgContainer, the object@id attribute value of the root of each EPG subtree shall be exposed in the <objectIDs> element of the <Feature> element.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	IZOWC	N

NOTE: UPnP AV MediaServer control points can use the objectID to initiate a query starting from that subtree to obtain EPG data.

7.4.6.2.5

[GUIDELINE] If a UPnP AV MediaServer implements the CDS:FreeFormQuery action, it shall include a <Feature> element with the @name attribute equal to “FFQ” in the FeatureList output argument of the CDS:GetFeatureList action. The @version attribute of the same <Feature> element shall have a value of “1”. This <Feature> element shall contain one or more <ObjectID> elements. Each <ObjectID> element shall contain the @id property of a CDS container which satisfies both of the following conditions:

- The CDS:FreeFormQuery action is supported for the CDS container and each of its descendant containers
- The CDS:FreeFormQuery action is not supported for the parent of the CDS container

Additionally the @level attribute of each <ObjectID> element shall indicate the support level for XQuery requests for the container represented by that <ObjectID> element as defined in 7.4.6.7.

[ATTRIBUTES]

M	R	DMS	M-DMD	n/a	[46]	XXZDZ	N

NOTE: This guideline repeats the UPnP requirements for the FFQ <Feature>. In [46].

7.4.6.2.6

[GUIDELINE] If a UPnP AV MediaServer exposes EPG Program items that represent channelized content, it shall implement the DLNA Extended Tuner to expose the device's channel lineup as specified in 7.4.5.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46]	JEKB7	N

NOTE: The channel lineup provides UPnP AV MediaServer control points with the values which can be used to search the EPG Items using the channelID property. The channels are represented as a collection of videoBroadcastItems or audioBroadcast Items. These videoBroadcastItems and audioBroadcast items do not need to provide a URL in their <res>-elements. If the channelID@type of an EPG item is "NETWORK" then the channelID contains the actual URL of the content. If the channelID@type is not "NETWORK" then channelID contains a identifier that is resolved to a tuner item. Therefore, a device which implements the UPnP AV Media Server EPG Device Option does not need to stream content from a physical tuner. Such a device might not even have a physical tuner.

7.4.6.3 MM/EPG EPG Object Persistence Guidelines

7.4.6.3.1

[GENERAL] This defines the object persistence requirements for a UPnP AV MediaServer when implementing the optional EPG Server Device Option as defined in guideline 7.4.6.2.

7.4.6.3.2

[GUIDELINE] A UPnP AV MediaServer should use the same CDS Program Item (i.e. a CDS Program Item with the same @id property) to represent an EPG event throughout the lifetime of that EPG event, except when executing the Service Reset Procedure described in Section 2.3.7.1 of [46].

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	VVDED	N

NOTE: Some UPnP AV MediaServer implementations delete a CDS object representing an EPG event and re-create a new CDS object for that same event when an update is received from an EPG source. Such behavior makes it difficult for UPnP AV MediaServer control points to keep track of EPG events of interest, and is discouraged by DLNA.

7.4.6.3.3

[GUIDELINE] If a UPnP AV MediaServer implements the ScheduledRecording Device Option and indicates support for the cds EPG record class as defined in guideline 7.4.4.11, then it shall use the same CDS Program Item (i.e. a CDS Program Item with the same @id property) to represent an EPG event throughout the lifetime of that EPG event, except when executing the Service Reset Procedure described in Section 2.3.7.1 of [46].

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	58U9B	N

7.4.6.3.4

[GUIDELINE] If a UPnP AV MediaServer always uses the same CDS Program Item to represent an EPG event throughout the lifetime of that EPG event (except when executing the Service Reset Procedure described in Section 2.3.7.1 of [46]), then it shall use the <dlna:X_DLNACAP> element (as a child of the <device> element that represents the UPnP AV MediaServer) in the device description document and include the Capability ID "epg-object-persistence" in the element's comma-separated value list.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [49]	JGDW8	N

7.4.6.4 MM/EPG EPG Controller Definition

7.4.6.4.1

[GUIDELINE] A DLNA device class may implement the EPG Controller Device Capability.

[ATTRIBUTES]

O	A	DMS DMP DMR DMC DMPr	M-DMS M-DMP M-DMD M-DMU	n/a	n/a	BG2YK	N

7.4.6.4.2

[GUIDELINE] An EPG Controller shall implement a UPnP AV MediaServer control point for interacting with a ContentDirectory service on a DMS or M-DMS.

[ATTRIBUTES]

M	R	+EPG+	n/a	n/a	[46] [49]	DKIL9	N

NOTE: This guideline indicates that an EPG Controller Device Capability will use a UPnP control point that controls a UPnP AV MediaServer for browsing content

7.4.6.5 MM/EPG Mandatory EPG Program Item Properties

7.4.6.5.1 MM/EPG General

This defines the mandatory properties that a UPnP AV MediaServer needs to expose for EPG Program Items.

7.4.6.5.2 MM/EPG upnp:class Property

7.4.6.5.2.1

[GUIDELINE] A UPnP AV MediaServer shall utilize the object.item.epgItem class or a derived class thereof for all EPG Program Items exposed in an EPG Container. The specific value of the upnp:class property shall conform to the requirements defined in 7.4.6.5.2.2 to 7.4.6.5.2.4.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	426J3	N

NOTE: All EPG Program Items will be derived from the object.item.epgItem base class.

7.4.6.5.2.2

[GUIDELINE] If a UPnP AV MediaServer is unable to determine the DLNA Media Class to which the Content Binary associated with the EPG Program Item conforms, it shall set the value of the upnp:class property to object.item.epgItem

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	7VCPH	N

NOTE: A UPnP AV MediaServer might not be able to determine the Media Format of the Content Binary associated with an EPG Program Item when that EPG Program Item is initially exposed, or even during the entire lifespan of the EPG Program Item. In that case, a UPnP AV MediaServer will set the upnp:class value to object.item.epgItem, and will not use one of the media format-specific derived classes. If the Media Format becomes known during the lifespan of the EPG Program Item, the UPnP AV MediaServer will update the value of the upnp:class property as specified in 7.4.6.5.2.3 and 7.4.6.5.2.4.

7.4.6.5.2.3

[GUIDELINE] If a UPnP AV MediaServer determines that the Content Binary associated with an EPG Program Item conforms to the DLNA AV Media Class, the value of the upnp:class property shall be set to object.item.epgItem.videoProgram.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	3CGXD	N

NOTE: A UPnP AV MediaServer will utilize the object.item.epgItem.videoProgram class for all EPG Program Items whose associated Content Binaries conform to the DLNA AV Media Class. It is not required that the DLNA Media Format Profile ID be exposed in the <res> element of the EPG Program Item.

7.4.6.5.2.4

[GUIDELINE] If a UPnP AV MediaServer determines that the Content Binary associated with an EPG Program Item conforms to the DLNA Audio Only Media Class, the value of the upnp:class property shall be set to object.item.epgItem.audioProgram.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	8WDPK	N

NOTE: A UPnP AV MediaServer will utilize the object.item.epgItem.audioProgram class for all EPG Program Items whose associated Content Binaries conform to the DLNA Audio Only Media Class.

7.4.6.5.2.5

[GUIDELINE] If a UPnP AV MediaServer determines that the Content Binary associated with an EPG Program Item does not conform to a DLNA Media Format Profile in the AV or Audio Only Media Classes, then it shall not expose the dlna.org_PN parameter in the 4th Field of the protocolInfo property.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	N6QVH	N

NOTE: In order to support the DLNA mission of interoperability, EPG Program Items whose content is not currently supported by the DLNA ecosystem will be not be identified as DLNA content.

7.4.6.5.2.6

[GUIDELINE] If a UPnP AV MediaServer determines that the Content Binary associated with an EPG Program Item conforms to a DLNA Media Format Profile in the AV or Audio Only Media Classes, then it should expose the dlna.org_PN parameter in the 4th Field of the protocolInfo property.

[ATTRIBUTES]

S	A	DMS	n/a	n/a	[46]	UHE9C	N

7.4.6.5.3 MM/EPG dc:title Property

7.4.6.5.3.1

[GUIDELINE] A UPnP AV MediaServer shall provide a value for the dc:title property that identifies the EPG Program Item.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	TDXMQ	N

NOTE: The value of the dc:title property will represent the program or title of the EPG Program Item.

7.4.6.5.3.2

[GUIDELINE] If a UPnP AV MediaServer exposes an EPG Program Item that does not have a DLNA Recognized Metadata Source, the value of dc:title should be set to best represent a user-friendly title for the contents represented by the EPG Program Item.

[ATTRIBUTES]

S	A	DMS	n/a	n/a	[46]	4FBKE	N

7.4.6.5.3.3

[GUIDELINE] If the EPG Program Item metadata source is OpenEPG, the value of the dc:title property shall be set to the value of the highest priority metadata (as listed below in order of decreasing priority) available for that EPG Program Item in the source metadata.

- Content.ContentTitle
- ContentServiceSource.Name
- ContentService.ContentServiceMapping.Channel
- ContentService.ContentServiceMapping.URLSource

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[106]	LY378	N

NOTE: At least one of the above-mentioned OpenEPG metadata elements will be present so that there will always be a value available to populate the dc:title property.

7.4.6.5.3.4

[GUIDELINE] If the EPG Program Item metadata source is TV-Anytime, the value of the dc:title property shall be set to the value of the highest priority metadata (as listed below in order of decreasing priority) available for that EPG Program Item in the source metadata.

- ProgramLocation.InstanceDescription.Title
- ProgramInformation.BasicDescription.Title

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[62]	IWD2N	N

NOTE: At least one of the above-mentioned TV-Anytime metadata elements will be present so that there will always be a value available to populate the dc:title property.

7.4.6.5.3.5

[GUIDELINE] If the EPG Program Item metadata source is DVB-SI, the value of the dc:title property shall be set to the value of the highest priority metadata (as listed below in order of decreasing priority) available for that EPG Program Item in the source metadata.

- short_event_descriptor.event_name
- service_descriptor.service_name

If neither the event_name nor the service_name are available, a vendor defined value shall be assigned.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[63]	LADLI	N

NOTE: At least one of the above-mentioned DVB-SI metadata elements will be present so that there will always be a value available to populate the dc:title property.

7.4.6.5.4 MM/EPG upnp:longDescription Property

7.4.6.5.4.1

[GUIDELINE] If a detailed description of the EPG Program Item is available, a UPnP AV MediaServer shall expose the upnp:longDescription property.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	XR6OP	N

NOTE: The value of the upnp:longDescription property provides a summary or synopsis for the EPG Program Item. If a suitable source for this property is not present in the source metadata, this property is omitted in the EPG Program Item.

7.4.6.5.4.2

[GUIDELINE] If a UPnP AV MediaServer exposes an EPG Program Item that does not have a DLNA Recognized Metadata Source, the value of the upnp:longDescription property shall provide the most descriptive summary of the EPG Program Item contents.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	74NJM	N

7.4.6.5.4.3

[GUIDELINE] If the EPG Program Item metadata source is OpenEPG, the value of the upnp:longDescription property shall be set to the value of the highest priority metadata (as listed below in order of decreasing priority) available for that EPG Program Item in the source metadata.

- Content.LongDescription
- Content.ShortDescription

If neither of these metadata elements are present in the source metadata, then the upnp:longDescription property shall be omitted from the EPG Program Item.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[106]	LC78K	N

7.4.6.5.4.4

[GUIDELINE] If the EPG Program Item metadata source is TV-Anytime, the value of the upnp:longDescription property shall be set to the value of the highest priority metadata (as listed below in order of decreasing priority) available for that EPG Program Item in the source metadata.

- (ProgramLocation.)InstanceDescription.Synopsis
- ProgramInformation.BasicDescription.Synopsis

If neither of these metadata elements are present in the source metadata, then the upnp:longDescription property shall be omitted from the EPG Program Item.

If multiple descriptions with different length attributes (“short”, “medium”, or “long”) exist in the source metadata, the instance with the highest length attribute (“long” > “medium” > “short”) shall be assigned to the upnp:longDescription property.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[62]	DBUAV	N

7.4.6.5.4.5

[GUIDELINE] If the EPG Program Item metadata source is DVB-SI, the value of the upnp:longDescription property shall be set to the value of the metadata listed below for that EPG Program Item in the source metadata.

- short.event.descriptor.text_char

If this metadata element is not present in the source metadata, then the upnp:longDescription property shall be omitted from the EPG Program Item.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[63]	JMW5D	N

7.4.6.5.5 MM/EPG dc:language Property

7.4.6.5.5.1

[GUIDELINE] If a list of languages is available (as different audio tracks of the EPG Program Item), each of these languages shall be exposed using an instance of the multi-valued dc:language property. Languages shall be represented in conformance with [64], for example, “en-US”.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [64]	Q477D	N

NOTE: The dc:language property is a multi-valued property allowing all languages to be represented (as separate instances of the dc:language property) when an EPG Program Item has multiple audio tracks, each in a different language.

7.4.6.5.5.2

[GUIDELINE] If a UPnP AV MediaServer exposes an EPG Program Item which does not have a DLNA Recognized Metadata Source, the EPG Program Item shall provide an instance of the dc:language property, set to the appropriate value, for each of the languages of the audio tracks of the associated Content Binary.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	V2ALV	N

7.4.6.5.5.3

[GUIDELINE] If the EPG Program Item metadata source is OpenEPG, then the following applies (in order of decreasing priority):

- If the OpenEPG Event.AudioLanguages metadata is present, a dc:language property instance shall be exposed and appropriately populated for each of the languages available in the OpenEPG Event.AudioLanguages metadata.
- If the OpenEPG Content.PrimaryLanguage is present, an instance of the dc:language property shall be exposed and appropriately populated with that value. In addition, for each Content.Alternative.Language that is present, an instance of the dc:language property shall be exposed and appropriately populated with each respective value.
- If the DistributionNetwork.ContentService.ContentServiceMapping.ServiceAudioLanguages

metadata is present, a dc:language property instance shall be exposed and appropriately populated for each of the languages available in the OpenEPG DistributionNetwork.ContentService.ContentServiceMapping.ServiceAudioLanguages metadata.

In this guideline, “appropriately populated” means a conversion from [65] language code used in the OpenEPG metadata to language representations in conformance with [64] is required.

If none of these metadata elements are present in the source metadata the dc:language, then property shall be omitted from the EPG Program Item.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[106] [65] [64]	58A9O	N

7.4.6.5.5.4

[GUIDELINE] If the EPG Program Item metadata source is TV-Anytime, then the following applies (in order of decreasing priority):

- If one or more TV-Anytime (ProgramLocation.)InstanceDescription.AVAttributes.AudioAttributes.AudioLanguage metadata elements are present, a dc:language property instance shall be exposed and appropriately populated for each of the languages available in the TV-Anytime ProgramLocation.)InstanceDescription.AVAttributes.AudioAttributes.AudioLanguage metadata elements. This requires a conversion from the ISO_639-1 language code, used in the TV-Anytime metadata to language representations in conformance with [64].
- If one or more TV-Anytime ProgramInformation.AVAttributes.AudioAttributes.AudioLanguage metadata elements are present, an instance of the dc:language property shall be exposed and appropriately populated for each of the TV-Anytime BasicDescription.AVAttributes.AudioAttributes.AudioLanguage metadata elements. This requires a conversion from the ISO_639-1 language code, used in the TV-Anytime metadata to language representations in conformance with [64].
- If the TV-Anytime ProgramInformation.BasicDescription.Language metadata is present, an instance of the dc:language property shall be exposed and populated with that value. This requires a conversion from the ISO_639-1 language code, used in the TV-Anytime metadata to language representations in conformance with [64].
- If the TV-Anytime ServiceInformation.ServiceLanguage metadata is present, a dc:language property instance shall be exposed and populated with that value. This requires a conversion from the ISO_639-1 language code, used in the TV-Anytime metadata to language representations in conformance with [64].

In this guideline, “appropriately populated” means a conversion from [65] language code used in the TV-Anytime metadata to language representations in conformance with [64] is required.

If none of these metadata elements are present in the source metadata the dc:language property shall be omitted from the EPG Program Item.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[62] [65]	F5TY7	N

7.4.6.5.5.5

[GUIDELINE] If the EPG Program Item metadata source is DVB-SI, then the following applies (in order of decreasing priority):

- If one or more DVB-SI event_information_section.component_descriptor metadata elements are present, a dc:language property instance shall be exposed and appropriately populated for each of the DVB-SI event_information_section.component_descriptor metadata elements. This requires a conversion from the ISO_639 language code, used in the DVB-SI metadata to language representations in conformance with [64].
- If one or more DVB-SI service_description_section.component_descriptor metadata elements are present, a dc:language property instance shall be exposed and appropriately populated for each of the DVB-SI service_description_section.component_descriptor metadata elements. This requires a conversion from the ISO_639 language code, used in the DVB-SI metadata to language representations in conformance with [64].

In this guideline, “appropriately populated” means a conversion from [65] language code used in the DVB-SI metadata to language representations in conformance with [64] is required.

If none of these metadata elements are present in the source metadata the dc:language property shall be omitted from the EPG Program Item.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[63] [65]	P2K7P	N

7.4.6.5.6 MM/EPG upnp:rating Property

7.4.6.5.6.1

[GUIDELINE] If Content Rating information is available for an audioProgram or videoProgram EPG Program Item, it shall be exposed using the upnp:rating property. The property shall include the upnp:rating@type property. DLNA recognized rating systems are listed in Annex L. When metadata source systems contain rating information that conforms to one of the DLNA recognized systems defined in Annex L the upnp:rating@type property shall be populated with a Domain name from Annex L and the upnp:rating property shall be populated with a rating value from Annex L that corresponds to the domain in the upnp:rating@type property If the source metadata does not include Content Rating information, the upnp:rating property shall be omitted from the EPG Program Item.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [RFC 3066]	JMC5Q	N

7.4.6.5.6.2

[GUIDELINE] If the rating domain exposed by a metadata schema is one of the systems listed in Annex L, then the properties shall be filled out as follows:

- If the system makes a parental rating available, and the <domain> is listed in the “Domain” column of Annex L then the rating@type property shall be <domain>
- If the system makes a parental rating available, the rating property shall be rating

  where rating is one of the valid ratings listed in the “Valid Ratings” column of Annex L for the appropriate <domain>.
- If the rating specified in the system has an age equivalence, the device shall expose a second rating@equivalentAge property with the value of age.

  where age is the age listed in the “Age Equivalence” column of Annex L for the appropriate <domain>.
- If the rating also include advice, the device should expose a rating@advice property with the value of advice.

where advice is a comma separated list of all the advice values that apply to this rating, as listed in the “Valid Advice” column of Annex L for the appropriate <domain>.

Example :

```
<rating type=”ACMA.GOV.ORG” equivalentAge =”15” advice=”V,L,S”>MA15+</rating>
```

[Attributes]

M	A	DMS	n/a	n/a	[46]	V9QX3	N

NOTE: the age column is intended to be used programmatically

7.4.6.5.6.3

[Guideline] If there is more than one rating for an individual rating authority then only the most restrictive rating should be exposed.

[Attributes]

S	A	DMS	n/a	n/a	[46]	AHP4I	N

7.4.6.5.6.4

[Guideline] If the rating system does not match any of the ratings systems listed in Annex L, but the rating system can be recognized by a unique and valid ICANN domain, then the device shall expose the rating system using the rules in 7.4.6.5.6.2 with a <domain> value that is a valid ICANN domain. The values of rating, equivalentAge and advice are set to any values that are specified for that domain, where the valid values are outside the scope of these DLNA guidelines.guidelines. The rating@type property shall be:

<domain> + “_PR” or <domain> + “_MA”

[Attributes]

M	A	DMS	n/a	n/a	[46]	A23TJ	N

7.4.6.5.6.5

[Guideline] If the rating system does not match any of the ratings systems listed in Annex L, and the rating system cannot be recognized by a unique and valid ICANN domain, then the device shall use the value “DLNA.ORG_UNKNOWN_PR:<rating system>” as the upnp:rating@type value.

[Attributes]

M	A	DMS	n/a	n/a	[46]	83AX3	N

NOTE: the age column is intended to be used programmatically

7.4.6.5.6.6

[Guideline] If the value of rating@type is one of the following:

- “cbsc.ca/english”
- “cbsc.ca/french”
- “mpaa.org”
- “kbc.go.kr”
- “gio.gov.tw”

The rating and rating@advice property should be exposed as defined in [107].

[ATTRIBUTES]

S	A	DMS	n/a	n/a	[46] [107]	N66VU	N

NOTE: In order to program the V-Chip, content will maintain its CEA-766 rating. This CEA-766 rating requires that all advice fields be maintained in the EPG, so that they can be translated back to the appropriate V-Chip bits. For more information, refer to [107].

7.4.6.5.6.7

[GUIDELINE] If the rating authority is listed in Annex L and the <rating> property value is a valid value in the Valid Ratings column for that domain then if the rating@equivalentAge property is empty, the device shall use the associated Age Equivalence value from Annex L as the value of rating@equivalentAge.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	Q6UBE	N

7.4.6.5.6.8

[GUIDELINE] If the rating authority is listed in Annex L and the rating@equivalentAge value is a valid value in the Age Equivalent column for that domain then if the rating property is empty, the device shall use the associated rating value from Annex L as the value of the rating property.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	8YGTV	N

7.4.6.5.6.9

[GUIDELINE] If the EPG Program Item metadata source is OpenEPG, then the following applies (in order of decreasing priority):

- Rating metadata in OpenEPG is found in three places. It may be associated with the Event, the Content, or the Service. If rating data is found in more than one of these elements associated with a single Event only the most specific metadata should be exposed. If there is Rating metadata in the Event object then the Event Rating metadata shall be used. If there is no metadata in the Event and there is Rating metadata in the Content object then the Content object Rating metadata shall be used. If neither the Event nor the Content object has Rating metadata then the rating metadata in the service object shall be used. Once the source of the rating metadata has been determined by the above rules then the mapping is done according to the following rules where <OpenEPGObject> is one of Event.EventContentAdvisory, Content.ContentAdvisory, or DistributionNetwork.ContentService.ContentServiceMapping.ContentAdvisory as determined above.
- If the OpenEPG RatingAuthority metadata conforms to one of the DLNA recognized rating authorities, in Annex L , then the upnp:rating@type property instance shall be exposed and populated as follows The <domain> parameter shall be set to the <OpenEPGObject>.RatingAuthority metadata value as described in 7.4.6.5.6.2 through 7.4.6.5.6.6.
    - upnp:rating@type = <domain>
- If the OpenEPG RatingAuthority metadata does not conform to one of the DLNA recognized rating authorities but a valid ICANN domain name can be established that uniquely identifies that rating authority, a upnp:rating@type property instance shall be exposed and the following syntax and assignment shall be used:
    - upnp:rating@type = <OpenEPGObject>.RatingAuthority + "_PR"

- If the OpenEPG RatingAuthority metadata does not conform to one of the DLNA recognized rating authorities and no valid ICANN domain name can be established that uniquely identifies that rating authority, a upnp:rating@type property instance shall be exposed and the following syntax and assignment shall be used:
  - upnp:rating@type = “DLNA.ORG_UNKNOWN_PR:” + <OpenEPGObject>.RatingAuthority
- If the OpenEPG RatingAuthority metadata conforms to one of the DLNA recognized rating authorities, in Annex L, then the upnp:rating property shall be exposed and populated as follows. The <OpenEPGObject>.ParentalRating metadata value shall be mapped to the appropriate Rating Value in Annex L.
  - upnp:rating = mapped Rating Value
- If the OpenEPG RatingAuthority metadata does not conform to one of the DLNA recognized rating authorities in Annex L, then the upnp:rating shall be set to the ParentalRating value
  - upnp:rating =<OpenEPGObject>.ParentalRating
- If the OpenEPG <OpenEPGObject>.ContentAdvisory metadata is present and contains MinimumAge then, a upnp:rating@ equivalentAge instance shall be exposed and assigned the minimum age value.
  - upnp:rating@equivalentAge = <OpenEPGObject>.MinimumAge

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[106]	UNCGW	N

7.4.6.5.6.10

[GUIDELINE] If the EPG Program Item metadata source is TV-Anytime, then the following applies (in order of decreasing priority):

- If the tva:TVAMain/tva:ProgramDescription/tva:GroupInformationTable/tva:GroupInformation/tva:BasicDescription/tva:ParentalGuidance metadata is present and indicates that the Program Item conforms to one of the DLNA recognized rating authorities listed in Annex L, then a upnp:rating and upnp:rating@type property pair instance shall be exposed and appropriately populated: the rating@type property shall be set to the DOMAIN value in Annex L that maps to the /tva:TVAMain/tva:ProgramDescription/tva:GroupInformationTable/tva:GroupInformation/tva:BasicDescription/tva:ParentalGuidance/mpeg7:Region metadata value and the <rating> property shall be set to the Annex L “Valid Ratings” value that maps to the tva:TVAMain/tva:ProgramDescription/tva:GroupInformationTable/tva:GroupInformation/tva:BasicDescription/tva:ParentalGuidance.ParentalRating metadata value
- If the TV-Anytime BasicDescription/tva:ParentalGuidance/mpeg7:ParentalRating metadata is present and indicates that the Program Item does not conform to one of the DLNA recognized rating authorities but a valid ICANN domain name can be established that uniquely identifies that rating authority, then a upnp:rating and upnp:rating@type property pair instance shall be exposed and the following syntax and assignments shall be used:
  - upnp:rating = /tva:TVAMain/tva:ProgramDescription/tva:GroupInformationTable/tva:GroupInformation/tva:BasicDescription/tva:ParentalGuidance/mpeg7:ParentalRating
  - upnp:rating@type = <valid ICANN domain name> + “_PR”
- If the TV-Anytime BasicDescription.ParentalGuidance .ParentalRating metadata is present and indicates that the Program Item does not conform to one of the DLNA recognized rating authorities and no valid ICANN domain name can be established that uniquely identifies that rating authority, then a upnp:rating and upnp:rating@type property pair instance shall be exposed and the following syntax and assignments shall be used:

  - upnp:rating = /tva:TVAMain/tva:ProgramDescription/tva:GroupInformationTable/tva:GroupInformation /tva:BasicDescription/tva:ParentalGuidance/mpeg7:ParentalRating
  - upnp:rating@type = “DLNA.ORG_UNKNOWN_PR:” + /tva:TVAMain/tva:ProgramDescription/tva:GroupInformationTable/tva:GroupInformation /tva:BasicDescription/tva:ParentalGuidance/mpeg7:Region
- If the TV-Anytime BasicDescription.ParentalGuidance metadata is present and contains MinimumAge metadata and a valid ICANN domain name can be established that uniquely identifies that rating authority, then a upnp:rating and upnp:rating@type property pair instance shall be exposed and the following syntax and assignments shall be used:
  - upnp:rating@equivalentAge = /tva:TVAMain/tva:ProgramDescription/tva:GroupInformationTable/tva:GroupInformation /tva:BasicDescription/tva:ParentalGuidance/mpeg7:MinimumAge
  - upnp:rating = /tva:TVAMain/tva:ProgramDescription/tva:GroupInformationTable/tva:GroupInformation /tva:BasicDescription/tva:ParentalGuidance/mpeg7:MinimumAge
  - upnp:rating@type = < domain name> + “_MA”
- If the TV-Anytime tva:TVAMain/tva:ProgramDescription/tva:GroupInformationTable/tva:GroupInformation/tva: BasicDescription/tva:ParentalGuidance metadata is present and contains MinimumAge metadata and indicates that the Program Item conforms to one of the DLNA recognized rating authorities listed in Annex L, then a upnp:rating and upnp:rating@type property pair instance shall be exposed and the following syntax and assignments shall be used:
  - upnp:rating@equivalentAge = /tva:TVAMain/tva:ProgramDescription/tva:GroupInformationTable/tva:GroupInformation /tva:BasicDescription/tva:ParentalGuidance/mpeg7:MinimumAge
  - upnp:rating@type = <domain name> (from Annex L)
- If the TV-Anytime BasicDescription.ParentalGuidance metadata is present and contains MinimumAge metadata and no valid ICANN domain name can be established that uniquely identifies that rating authority, then a upnp:rating and upnp:rating@type property pair instance shall be exposed and the following syntax and assignments shall be used:
  - upnp:rating@equivalentAge = /tva:TVAMain/tva:ProgramDescription/tva:GroupInformationTable/tva:GroupInformation /tva:BasicDescription/tva:ParentalGuidance/mpeg7:MinimumAge
  - upnp:rating = /tva:TVAMain/tva:ProgramDescription/tva:GroupInformationTable/tva:GroupInformation /tva:BasicDescription/tva:ParentalGuidance/mpeg7:MinimumAge
  - upnp:rating@type = “DLNA.ORG_UNKNOWN_MA:” + /tva:TVAMain/tva:ProgramDescription/tva:GroupInformationTable/tva:GroupInformation /tva:BasicDescription/tva:ParentalGuidance/mpeg7:Region

If none of these metadata elements are present in the source metadata, then the upnp:rating property shall be omitted from the EPG Program Item.

[Attributes]

M	A	DMS	n/a	n/a	[62]	F7NL9	N

7.4.6.5.6.11

[Guideline] If the EPG Program Item metadata source is DVB-SI, then the following applies (in order of decreasing priority):

- If the DVB-SI event_information_section.parental_rating_descriptor metadata is present, then a upnp:rating and upnp:rating@type property pair instance shall be exposed. The 24-

bit country code metadata shall be converted according to [68] and the resulting <Country> code shall be used as follows:

- If a rating system in Annex L can be determined from the country code then
  - upnp:rating@type = <domain>
  - If the value of the <Rating> metadata is less than 16 then
    - <Rating> = <Rating> + 3
    - upnp:rating@equivalentAge= <Rating>
    - upnp:rating = rating that matches the equivalentAge value
- If the rating can be mapped to a valid rating value for the specified domain then
  - rating = rating value from Annex L
  - upnp:rating = rating
- If a rating system in Annex L cannot be determined from the country code then the value shall be used in the assignments below. Likewise, the 8-bit <Rating> code shall be used as follows:
  - If the value of the <Rating> metadata is less than 16 then
    - <Rating> = <Rating> + 3
    - upnp:rating@equivalentAge= <Rating>
    - upnp:rating@= <Rating>
    - upnp:rating@type = “DLNA.ORG_UNKNOWN” + “_MA:” + <Country>
  - else
    - upnp:rating = “Broadcaster Defined:” + <Rating>
    - upnp:rating@type = “DLNA.ORG_UNKNOWN” +”_PR:” + <Country>

If the metadata element is not present in the source metadata, then the upnp:rating property shall be omitted from the EPG Program Item.

[Attributes]

M	A	DMS	n/a	n/a	[63] [68]	LIPFO	N

7.4.6.5.7 MM/EPG upnp:channelID Property

7.4.6.5.7.1

[Guideline] An EPG Program Item CDS object may expose one or more instances of the upnp:channelID property.

[Attributes]

O	L	DMS	M-DMS	n/a	[46]	F953R	N

NOTE: The upnp:channelID property is used across the SRS and EPG system usages as the common identifier for channels.>

7.4.6.5.7.2

[Guideline] If a UPnP AV MediaServer exposes one or more instances of the upnp:channelID property, each shall have a value that identifies the channel source of the EPG Program Item. Each upnp:channelID property shall include the upnp:channelID@type property. The upnp:channelID property is multi-valued so that different formats can be used to identify a particular channel. When multiple instances of the upnp:channelID property are exposed, they shall refer to the same channel item in the CDS channel lineup. The value of the upnp:channelID@type property shall conform to the guidelines in 7.4.6.5.7.3 to 7.4.6.5.7.11.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	WW3W8	N

7.4.6.5.7.3

[GUIDELINE] If a UPnP AV MediaServer exposes a vendor-specific channel type, then the upnp:channelID@type value shall incorporate a valid ICANN registered domain name that uniquely identifies the vendor.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	BFQIE	N

7.4.6.5.7.4

[GUIDELINE] If a UPnP AV MediaServer exposes an EPG Program Item with a upnp:channelID@type value of “ANALOG”, then the value of the upnp:channelID property shall be a positive integer.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	GZNAJ	N

NOTE This enumerated type is potentially misleading as the source is not required to be an analog broadcast source. Vendors are encouraged to use this enumerated type for any channel items which are represented by a channel identifier consisting of a single integer.

7.4.6.5.7.5

[GUIDELINE] A UPnP AV MediaServer shall not use the upnp:channelID@type value of “ANALOG” to represent a major / minor channel number using “aliasing” (such as multiplying the major channel number by 1000 then adding the minor channel number). Channel numbers with a major / minor channel number shall be exposed using the upnp:channelID@type value of “DIGITAL” as defined in 7.4.6.5.7.6.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	E7I45	N

NOTE The terms analog and digital do not refer to the broadcast system. The meaning is only distinguishing the numbering system.

7.4.6.5.7.6

[GUIDELINE] If a UPnP AV Media Server exposes an EPG Program Item with a upnp:channelID@type value of “DIGITAL”, then the value of the upnp:channelID property shall consist of a “major” and “minor” channel number pair in the format

- <Major Channel Number>,<Minor Channel Number>

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	LI5G3	N

NOTE Implementers need to note that UPnP utilizes a comma to separate the major and minor channel numbers while they are often separated with a period.

7.4.6.5.7.7

[GUIDELINE] If a UPnP AV MediaServer exposes an EPG Program Item with a upnp:channelID@type value of “SI”, then the value of the upnp:channelID property shall consist of a Service Information Triplet in the format

- The <Network ID> term is a non-negative numerical value with a range of 0 to 0xFFFF. The value is network specific and shall be represented as a decimal or hexadecimal value. The value shall be omitted (i.e. empty string) to indicate an unknown <network ID> term.
- The <Transport Stream ID> term is a non-negative numerical value with a range of 0 to 0xFFFF. The value is network specific and shall be represented as a decimal or hexadecimal value. The value shall be omitted (i.e. empty string) to indicate an unknown <Transport Stream ID> term.
- The <Service ID> term is a non-negative numerical value with a range of 0 to 0xFFFF. The value is network specific and can be represented as a decimal or hexadecimal value.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	3C6H7	N

NOTE This provides clarification for the values contained within the upnp:channelID property for the "SI" type that is lacking in [46]. Examples of valid values for the upnp:channelID are as follows:

- <upnp:channelID type="SI">0x1234,0xFEDC,0x0102</upnp:channelID>
- <upnp:channelID type="SI">12345,23456,32109</upnp:channelID>
- <upnp:channelID type="SI">,1,0x0102</upnp:channelID>
- <upnp:channelID type="SI>,,0x0102</upnp:channelID>

7.4.6.5.7.8

[GUIDELINE] If a UPnP AV MediaServer exposes an EPG Program Item with a upnp:channelID@type value of "NETWORK", then the value of the upnp:channelID property shall consist of a properly formatted URI.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	CDIET	N

NOTE The NETWORK type is utilized for non-traditional broadcast sources which present content to the consumer using a "channel metaphor".

7.4.6.5.7.9

[GUIDELINE] If the EPG Program Item metadata source is OpenEPG, then the values of the upnp:channelID and the upnp:channelID@type properties shall be obtained from the metadata as defined below (in order of decreasing priority):

- If the OpenEPG DistributionNetwork.ContentService.ContentServiceMapping is present and contains both the Channel and MinorChannel metadata elements, then:
  - upnp:channelID = Channel + "," + MinorChannel
  - upnp:channelID@type = "DIGITAL"
- If the OpenEPG DistributionNetwork.ContentService.ContentServiceMapping is present and contains only the Channel metadata element, then:
  - upnp:channelID = Channel
  - upnp:channelID@type = "ANALOG"
- If the OpenEPG DistributionNetwork.ContentService.ContentServiceMapping is present and contains the URLSource metadata element, then:
  - upnp:channelID = URLSource
  - upnp:channelID@type = "NETWORK"

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[106]	X6ZPW	N

7.4.6.5.7.10

[GUIDELINE] If the EPG Program Item metadata source is TV-Anytime, then the values of the upnp:channelID and the upnp:channelID@type properties shall be obtained from the metadata as defined below (in order of decreasing priority):

- If the TV ProgramLocationTable.ProgramURL is present and its value starts with the string “dvb://”, then:
  - upnp:channelID = <Network ID> + “,” + <Transport Stream ID> + “,” + <Service ID> as parsed from the ProgramURL metadata according to [62] and [67]
  - upnp:channelID@type = “SI”
- If the TV ProgramLocationTable.ProgramURL is present and its value starts with the string “PAL://”, “NTSC://”, or “SECAM://” then:
  - upnp:channelID = <channel> as parsed from the ProgramURL metadata according to [62] and [67]
  - upnp:channelID@type = “ANALOG”

If none of these properties are present in the source metadata the upnp:channelID property shall be omitted for the EPG Program Item.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[62] [67]	EAR7P	N

7.4.6.5.7.11

[GUIDELINE] If the EPG Program Item metadata source is DVB-SI, then the value of the upnp:channelID shall be obtained from the SDT table metadata as defined below.

- upnp:channelID = original_network_id + “,” + transport_stream_id + “,” + service_id
- upnp:channelID@type = “SI”

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[63]	TDN6K	N

7.4.6.5.7.12

[GUIDELINE] If a UPnP AV MediaServer exposes a Channelized EPG Program Item, then it shall include one or more instances of the upnp:channelID property.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	FBS7S	N

NOTE: The upnp:channelID property identifies the channel through which the Program Item will be delivered through for channelized content.

7.4.6.5.8 MM/EPG upnp:channelID@distriNetworkID Property

7.4.6.5.8.1

[GUIDELINE] A UPnP AV MediaServer EPG Program Item shall include the upnp:channelID@distriNetworkID property.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	W5EQB	N

NOTE: The upnp:channelID@distriNetworkID property identifies the Distribution Network from which the channel is sourced.

7.4.6.5.8.2

[Guideline] If a UPnP AV MediaServer exposes an EPG Program Item which does not have a DLNA Recognized Metadata Source, then the value of the upnp:channelID@distriNetworkID property shall reflect the distribution source of the channel.

[Attributes]

M	A	DMS	n/a	n/a	[46]	2PPZ2	N

NOTE: The upnp:channelID@distriNetworkID property is used by the user/application to determine what server/service will provide the content.

7.4.6.5.8.3

[Guideline] If the EPG Program Item metadata source is OpenEPG, then the value of the upnp:channelID@distriNetworkID property shall be set to the value of the OpenEPG DistributionNetwork.DistributionNetworkID metadata.

[Attributes]

M	A	DMS	n/a	n/a	[106]	LA35C	N

7.4.6.5.8.4

[Guideline] If the EPG Program Item metadata source is TV-Anytime, then the upnp:channelID@distriNetworkID property shall contain information that allows the user/application to determine the server/service providing the content identified by the EPG metadata."

[Attributes]

M	A	DMS	n/a	n/a	[62]	ANNC2	N

7.4.6.5.8.5

[Guideline] If the EPG Program Item metadata source is DVB-SI, then the value of the upnp:channelID@distriNetworkID property shall be set to the network_id of the NIT table

[Attributes]

M	A	DMS	n/a	n/a	[63]	J3BRD	N

7.4.6.5.9 MM/EPG upnp:channelID@distriNetworkName Property

7.4.6.5.9.1

[Guideline] A UPnP AV MediaServer EPG Program Item should include the upnp:channelID@distriNetworkName property.

[Attributes]

S	A	DMS	M-DMS	Na	[46]	NZF7S	N

NOTE: The upnp:channelID@distriNetworkName property identifies the Distribution Network from which the channel is sourced.

7.4.6.5.9.2

[GUIDELINE] If a UPnP AV MediaServer exposes an EPG Program Item which does not have a DLNA Recognized Metadata Source, the value of the upnp:channelID@distriNetworkName property should reflect the distribution source of the channel.

[ATTRIBUTES]

S	A	DMS	n/a	n/a	[46]	W7ETL	N

7.4.6.5.9.3

[GUIDELINE] If the EPG Program Item metadata source is OpenEPG, then the value of the upnp:channelID@distriNetworkName property shall be set to the value of the highest priority metadata (as listed below in order of decreasing priority) available for the specific EPG Program Item in the source metadata.

- DistributionNetwork.Name
- DistributionNetwork.DistributionNetworkID

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[106]	7XVTL	N

7.4.6.5.9.4

[GUIDELINE] If the EPG Program Item metadata source is DVB-SI, then the value of the upnp:channelID@distriNetworkName property shall be set to the:

- Network_name_descriptor in the NIT

If this property is not present in the source metadata the upnp:channelID@distriNetworkName property shall be omitted from the EPG Program Item.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[63]	HUM35	N

7.4.6.5.10 MM/EPG Time Related Properties

7.4.6.5.10.1

[GUIDELINE] An EPG Program Item should include the upnp:scheduledStartTime, upnp:scheduledStartTime@usage, upnp:scheduledEndTime, and upnp:scheduledDuration properties.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[46]	CGDZ5	N

NOTE: The upnp:scheduledStartTime property indicates the time the Program Item is scheduled to start or become available. The upnp:scheduledEndTime property indicates the time the Program Item is scheduled to end or cease to be available. The upnp:scheduledDuration property indicates the actual duration of the program content. Control points need to examine the upnp:scheduledStartTime@usage property to obtain the context of the time values.

7.4.6.5.10.2

[GUIDELINE] If an EPG Program Item exposes the upnp:scheduledStartTime or upnp:scheduledEndTime property, the property value shall represent local time.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	O3SSH	N

7.4.6.5.10.3

[GUIDELINE] If a UPnP AV MediaServer exposes an EPG Program Item which does not have a DLNA Recognized Metadata Source, then the values of the upnp:scheduledStartTime, upnp:scheduledStartTime@usage, upnp:scheduledEndTime, and upnp:scheduledDuration properties shall be set as follows:

- If the Program Item refers to scheduled program content, then the value of the upnp:scheduledStartTime property shall reflect the time the program content is scheduled to start. The value of the upnp:scheduledEndTime property shall reflect the time the program content is scheduled to end. The upnp:scheduledStartTime@usage property shall be set to “SCHEDULED_PROGRAM”. The value of the upnp:scheduledDuration property shall reflect the duration of the program content. One of the fields, upnp:scheduledDuration or upnp:scheduledEndTime, could be empty. Its value can be calculated from the other field and upnp:scheduledStartTime.
- If the Program Item refers to on-demand program content, then the value of the upnp:scheduledStartTime property shall indicate the beginning of the time window during which the on-demand content is available for consumption . The value of the upnp:scheduledEndTime property shall indicate the end of the time window during which the on-demand content is available for consumption. The upnp:scheduledStartTime@usage property shall be set to “ON_DEMAND”. The value of the upnp:scheduledDuration property shall reflect the duration of the program content and never the length of the time window during which the on-demand content is available for consumption.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[46]	3GTOF	N

7.4.6.5.10.4

[GUIDELINE] If the EPG Program Item metadata source is OpenEPG, then the values of the upnp:scheduledStartTime, upnp:scheduledStartTime@usage, upnp:scheduledEndTime, and upnp:scheduledDuration properties shall be set as follows:

- If the OpenEPG Event.StartTime is present, then:
  - upnp:scheduledStartTime = Event.StartTime
  - upnp:scheduledStartTime@usage = “SCHEDULED_PROGRAM”
- If the OpenEPG Event.EndTime metadata is present, then:
  - upnp:scheduledEndTime = Event.EndTime
  - upnp.scheduledDuration = Event.Duration or Event.EndTime – Event.StartTime
- If the OpenEPG Event.Duration metadata is present, then:
  - upnp:scheduledEndTime = Event.EndTime or Event.StartTime + Event.Duration
  - upnp.scheduledDuration = Event.Duration
- If the OpenEPG Event.VODStartTime and Event.VODEndTime metadata are present, then:
  - upnp:scheduledStartTime = Event.VODStartTime
  - upnp:scheduledStartTime@usage = “ON_DEMAND”
  - upnp:scheduledEndTime = Event.VODEndTime
  - upnp.scheduledDuration = Event.Duration
- If both OpenEPG Event.Duration and OpenEPG Event.scheduledEndTime are missing:
  - upnp:scheduledEndTime = OpenEPG Event.StartTime (of the following Event)

- upnp.scheduledDuration = OpenEPG Event.StartTime (of the following Event) - OpenEPG Event.StartTime (of this Event)

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[106]	2Z8FJ	N

NOTE: If one of the fields, Event.Duration or Event.scheduledEndTime, is empty its value needs to be calculated from the other field and Event.StartTime. If both are missing the Event.StartTime of the following event can be used as the Event.EndTime for this event.

7.4.6.5.10.5

[GUIDELINE] If the EPG Program Item metadata source is TV-Anytime, then the values of the upnp:scheduledStartTime, upnp:scheduledStartTime@usage, upnp:scheduledEndTime, and upnp:scheduledDuration properties shall be set or omitted as follows in 7.4.6.5.10.6 and 7.4.6.5.10.7.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[62]	Z2FIY	N

7.4.6.5.10.6

[GUIDELINE] If the EPG Program Item metadata source is TV-Anytime and the programURL is published with the schedule metadata and:

- It is a ScheduleEvent then:
  - upnp:scheduledStartTime = /tva:TVAMain/tva:ProgramDescription/tva:ProgramLocationTable/tva:Schedule/tva:ScheduleEvent/tva:PublishedStartTime
  - upnp:scheduledEndTime = /tva:TVAMain/tva:ProgramDescription/tva:ProgramLocationTable/tva:Schedule/tva:ScheduleEvent/tva:PublishedEndTime
  - upnp:scheduledDuration = /tva:TVAMain/tva:ProgramDescription/tva:ProgramLocationTable/tva:Schedule/tva:ScheduleEvent/tva:PublishedDuration
  - upnp:scheduledStartTime@usage property = “SCHEDULED_PROGRAM”
- It is a BroadcastEvent then:
  - upnp:scheduledStartTime = /tva:TVAMain/tva:ProgramDescription/tva:ProgramLocationTable/tva:BroadcastEvent/tva:PublishedStartTime
  - upnp:scheduledEndTime = /tva:TVAMain/tva:ProgramDescription/tva:ProgramLocationTable/tva:BroadcastEvent/tva:PublishedEndTime
  - upnp:scheduledDuration = tva:TVAMain/tva:ProgramDescription/tva:ProgramLocationTable/tva:BroadcastEvent/tva:PublishedDuration
  - upnp:scheduledStartTime@usage property = “SCHEDULED_PROGRAM”
- It is a OnDemandProgram then:
  - upnp:scheduledStartTime = /tva:TVAMain/tva:ProgramDescription/tva:ProgramLocationTable/tva:OnDemandProgram/tva:StartOfAvailability
  - upnp:scheduledEndTime = /tva:TVAMain/tva:ProgramDescription/tva:ProgramLocationTable/tva:OnDemandProgram/tva:EndOfAvailability

- upnp:scheduledDuration = /tva:TVAMain/tva:ProgramDescription/tva:ProgramLocationTable/tva:OnDemandProgram/tva:PublishedDuration
- upnp:scheduledStartTime@usage property = "ON DEMAND"

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[62]	G5BHR	N

7.4.6.5.10.7

[GUIDELINE] If the EPG Program Item metadata source is TV-Anytime and the programURL is the result of resolving a CRID into a locator then:

- upnp:scheduledStartTime@usage = /cr:ContentReferencingTable/cr:Result/cr:LocationsResult/cr:DecomposedLocator@mode (scheduled or ondemand) -> upnp:scheduledStartTime@usage
  - A broadcast event, which is a non-scheduled event (e.g. a news flash) is not concerned by CRID resolution
- upnp:scheduledStartTime = /cr:ContentReferencingTable/cr:Result/cr:LocationsResult/cr:DecomposedLocator@start ->
- upnp:scheduledDuration = /cr:ContentReferencingTable/cr:Result/cr:LocationsResult/cr:DecomposedLocator@duration
- upnp:scheduledEndTime = /cr:ContentReferencingTable/cr:Result/cr:LocationsResult/cr:DecomposedLocator@end

If neither of these properties are present in the source metadata, then the upnp:scheduledStartTime, upnp:scheduledStartTime@usage, upnp:scheduledEndTime, and upnp:scheduledDuration properties shall be omitted from the EPG Program Item.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[62]	IYW6R	N

NOTE: The programURL can be available from different sources. It is published with the schedule metadata and TV-Anytime addresses three types of programmes: Broadcast events (for instance an unscheduled news announcement), schedule events (or SCHEDULED PROGRAMME) and on-demand programmes. The mode upnp:scheduledStartTime@usage is derived from the root (inc. onDemand).

7.4.6.5.10.8

[GUIDELINE] If the EPG Program Item metadata source is DVB-SI, then the values of the upnp:scheduledStartTime, upnp:scheduledStartTime@usage, upnp:scheduledEndTime, and upnp:scheduledDuration properties shall be set as follows:

- upnp:scheduledStartTime = event.information.section.start_time +time_offset_section.local_time
- upnp:scheduledStartTime@usage = "SCHEDULED_PROGRAM"
- upnp:scheduledEndTime = event.information.section.start_time + event.information.section.duration +time_offset_section.local_time
- upnp:scheduledDuration = event.information.section.duration

If one or more of the properties are not present in the source metadata, then the upnp:scheduledStartTime, upnp:scheduledStartTime@usage, upnp:scheduledEndTime, and upnp:scheduledDuration properties shall be omitted for the EPG Program Item.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[63]	ZSIPA	N

7.4.6.5.11 MM/EPG @daylightSaving Properties

[GUIDELINE] A EPG Program Item shall include the upnp:scheduledStartTime@daylightSaving and upnp:scheduledEndTime@daylightSaving properties. Both properties shall have the same value and this property. The value of the property shall indicate whether the time values are expressed in Standard Time or Daylight Saving Time, and shall accurately reflect the application of Daylight Saving Time to the local time of the EPG Server.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46] [RFC 3066]	NHECC	N

NOTE: The upnp:scheduledStartTime property will be expressed in local time including the indication of Daylight Saving Time. When an EPG Server is in a location where Daylight Saving Time is not observed, this property is still needed and all times will indicate Standard Time.

7.4.6.6 MM/EPG Exposing Foreign Metadata

7.4.6.6.1

[GUIDELINE] The upnp:foreignMetadata::fmEmbeddedXML property of an EPG Item shall only contain information pertaining to the single program event that this EPG Item represents. The property value shall be a valid XML document in the format indicated by the upnp:foreignMetadata@type property.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	WJLQV	N

7.4.6.6.2

[GUIDELINE] For foreign Metadata from TVAnytime the @type property of a ::fmEmbeddedXML property in a CDS Event object shall have the value of “tv-anytime.org_1” to indicate TV-Anytime based metadata.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	EHTZM	N

7.4.6.6.3

[GUIDELINE] If the value of the @type property of a ::fmEmbeddedXML property in a CDS Event object is “tv-anytime.org_1”, then the ::TVAMain property shall include a ::ProgramInformationTable property which shall include a BasicDescription property for the current EPG Item. TV-Anytime information from different tables shall be integrated into a single <TVAMain> element.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	889NO	N

NOTE: An example of an EPGItem with TV-Anytime foreign metadata is provided below. It contains a single program description as well as a location description of this program.

```
<upnp:foreignMetadata type="tv-anytime.org_1">
   <upnp:fmId></upnp:fmId>
   <upnp:fmClass></upnp:fmClass>
   <upnp:fmProvider>dlna.examples.com</upnp:fmProvider>
   <upnp:fmBody xmlFlag="1" mimeType="text/xml">
      <upnp:fmEmbeddedXML>
```

```
          <TVAMain xmlns='urn:tva:metadata:2005' xmlns:mpeg7='urn:tva:mpeg7:2005'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xml:lang='en'>
              <ProgramDescription>
                  <ProgramInformationTable>
                      <ProgramInformation programId="crid://dlna.examples.com
/8730311156">
                          <BasicDescription>
                              <Title><Hot tech></Title>
                              <Synopsis length="short">Popular science program
explaining the wonders of DLNA home networks.</Synopsis>
                              <Genre href="urn:tva:metadata:cs:IntentionCS:2004:1.2">
                                  <Name>Information</Name>
                              </Genre>
                          </BasicDescription>
                          <AVAttributes>
                              <AudioAttributes>
                                  <NumOfChannels>2</NumOfChannels>
                              </AudioAttributes>
                              <VideoAttributes>
                                  <AspectRatio>16:9</AspectRatio>
                              </VideoAttributes>
                          </AVAttributes>
                      </ProgramInformation>
                  </ProgramInformationTable>
                  <ProgramLocationTable>
                      <Schedule serviceIDRef='HotSci' start='2007-06-01T00:00:00Z'
end='2007-06-01T23:59:59Z'>
                          <ScheduleEvent>
                          <Program
crid="crid://dlna.examples.com/8730311156"></Program>
    <ProgramURL>dvb://277f.ff00.2250;787@2007-06-01T21:00:00Z/PT01H45M</ProgramURL>
                          <PublishedStartTime>2007-061T21:00:00Z</PublishedStartTime>
                          <PublishedDuration>PT01H45M00S</PublishedDuration>
                          </ScheduleEvent>
                      </Schedule>
                  </ProgramLocationTable>
              </ProgramDescription>
          </TVAMain>
      </upnp:fmEmbeddedXML>
    </upnp:fmBody>
</upnp:foreignMetadata>
```

7.4.6.6.4

[GUIDELINE] If the value of the @type property of a ::fmEmbeddedXML property in a CDS Event object is “tv-anytime.org_1”, then the <ProgramDescription> element of ::TVAMain property may contain other tables having with information related to the current EPG Item (such as the ProgramLocationTable).

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[46]	C3FU7	N

7.4.6.6.5

[GUIDELINE] For foreign Metadata from OpenEPG the @type property of a ::fmEmbeddedXML property in a CDS Event object shall have the value of “openepg.org_v1” to indicate OpenEPG-based metadata.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	OFZPQ	N

7.4.6.7 MM/EPG Search Guidelines

7.4.6.7.1

[General] This defines the search requirements for UPnP AV MediaServers that implement the EPG Server Device Option. The mechanism to obtain EPG data from a server is based on search queries using the CDS:FreeFromQuery() action. A subset of the XQuery language is defined to allow for efficient implementations.

7.4.6.7.2

[Guideline] A UPnP AV MediaServer implementing the EPG Server Device Option shall implement the CDS:FreeFormQuery action.

[Attributes]

M	R	DMS	M-DMS	n/a	[46]	9TP5W	N

NOTE: UPnP AV MediaServer control points rely on FreeFormQuery searches to select EPGItems exposed by the CDS.

7.4.6.7.3

[Guideline] If the full XQuery language is supported for a certain sub-tree, its ObjectID@level attribute as defined in 7.4.6.2.5 shall be set to “0”. If a subset of XQuery is supported, the ObjectID@level shall be a comma separated value list containing values that identify the subset(s) being supported. The comma separated value list shall not contain the value “0”. The first value in the comma separated value list shall be “DLNA_EPG”.

[Attributes]

M	R	DMS	M-DMS	n/a	[46]	6ZJXJ	N

NOTE: Control points can use the ObjectID@level to check what level of XQuery complexity this server can handle. Vendors can define additional values to indicate support of additional XQuery subsets. The definition and usage of these vendor-defined values are out of scope.

7.4.6.7.4

[Guideline] If a UPnP AV MediaServer includes the @id attribute of a CDS container in the <ObjectIDs> element of the <Feature> element defined in 7.4.6.2.3 (i.e. the EPG, EPGDataOnly <Feature> element), then one of the <ObjectID> elements of the <Feature> element defined in 7.4.6.2.5 (i.e. the FFQ <Feature> element) shall contain the @id attribute of that CDS container or one of its ancestor containers

[Attributes]

M	R	DMS	M-DMS	n/a	[46]	EC7BH	N

NOTE: DLNA requires the CDS:FreeFormQuery action to be at least supported for all exposed EPG Containers. For example, when a UPnP AV MediaServer exposes two EPG trees, the ObjectIDs are listed in the result of the CDS:GetFeatureList action for the EPG feature. The FFQ <Feature> element indicates that CDS:FreeFormQuery is supported for these two EPG containers. This means that the FFQ <Feature> element can list these two ObjectIDs, one of the listed ObjectIDs plus an ancestor ObjectID, two different ancestors, or one common ancestor. An example of the latter case is a UPnP AV MediaServer which allows CDS:FreeFormQuery actions on all containers. In this case the FFQ <Feature> element will contain the root ObjectID.

7.4.6.7.5

[Guideline] A UPnP AV MediaServer that implements the EPG Server Device Option shall support the XQuery language subset defined by the following EBNF notation. This subset is referred to as the “DLNA_EPG” level XQuery.

This level of XQuery uses the constructor to produce a DIDL-Lite compliant document.

- Query ::= ConstructorSTag EnclosedExpression ConstructorETag

The DIDL-Lite fragment is specified as a constructor. The start tag shall have namespace declarations for the specified properties used in the query body.

- ConstructorSTag ::= ‘<DIDL-Lite’ DidlliteNSAttName DcNSAttName? UpnpNSAttName? OtherNSAttName* ‘>
- DidlliteNSAttName ::= 'xmlns= "urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/ "’
- DcNSAttName ::= ‘xmlns:dc=”http://purl.org/dc/elements/1.1/”’
- UpnpNSAttName ::= ’xmlns:upnp=“urn:schemas-upnp-org:metadata-1-0/upnp/“’
- OtherNSAttName ::= <Any other namespaces used in the query body shall be declared. See http://www.w3.org/TR/REC-xml-names/ for the syntax definitions.>

The main query body uses a Flwor expression. Like StartCount and RequestedCount arguments of a CDS::Browse action, an EPG Controller can limit the returned items to the specified range from the entire result with “fn:position”.

- EnclosedExpression ::= ‘{‘ SingleMainExpr ‘}’
- SingleMainExpr ::= FlworExpr | ‘(‘ FlworExpr ‘)’ ‘[‘ ‘fn:position()’ ‘=’ ‘(‘ Integer “to” Integer ‘)’ ‘]’
- FlworExpr ::= ForClause WhereClause? OrderByClause? ReturnClause

The For clause is static. Only items of epgItem or its derived class can be queried. DIDL-Lite is always specified as a root document even if the query is scoped in an EPG tree of the CDS. The UPnP AV MediaServer that implements the EPG Server Device Option shall limit the search scope within the subtree specified in the CDS::FreeFormQuery action. NoRef shall be specified in this level. This removes all duplicated epgItems from the result.

- ForClause ::= ‘for’ ‘$x’ ‘in’ ‘DIDL-Lite//item’ ‘[‘ EPGItem ‘AND’ NoRef ‘]’
- EPGItem ::= ‘fn:starts-with(upnp:class,’object.item.epgItem’)’
- NoRef ::= ’fn:not(fn:exists(@refID))’

An EPG Controller can request filtering of the epgItems out by time range, channel, title and long description. Channel line up can be structured per distribution network.

- WhereExpr ::= ‘where’ TimeExpr (‘AND’ ChannelExpr)? (‘AND’ StringExpr)?
- TimeExpr ::= ‘$x/upnp:scheduledStartTime’ ‘>=’ TimeLiteral ‘AND’ ‘$x/upnp:scheduledEndTime ‘ ‘<’ TimeLiteral
- TimeLiteral ::= <search value. see x.x.x.x for the syntax of timeliteral properties.>
- ChannelExpr ::= Channel | ‘(‘ ChannelList ‘)’ | ChannelDistr | ‘(‘ChannelDistriList ‘)’
- ChannelList ::= Channel (‘OR’ Channel)*
- ChannelDistriList ::= ChannelDistri (‘OR’ ChannelDistri)*
- Channel ::= ‘$x/upnp:channelID’ ‘=’ Identifier
- ChannelDistri ::= ‘$x/upnp:channelID/@distriNetworkID’ ‘=’ Identifier ‘AND’ (Channel | ‘(‘ ChannelList ‘)’)
- Identifier ::= <search target value>
- StringExpr ::= StrCmpANDList | ‘(‘StrCmpORList ‘)’
- StrCmpANDList ::= StrCmp (‘AND’ StrCmp)*
- StrCmpORList ::= StrCmp (‘OR’ StrCmp)*
- StrCmp ::= ‘fn:contains(‘ ‘$x/’ StrCmpProp ‘,’ ‘"’ Identifier ‘"’ ‘)’

- StrCmpProp ::= ‘dc:title’ | ‘upnp:longDescription’

An EPG Controller can request sorting of the result by start time and channel. Channel line up can be structured per distribution network. Only ascending order is allowed.

- OrderByClause ::= ‘order by’ SortList
- SortList ::= ((SortDistriNetworkID “,”)? SortChannelID “,”)? SortStartTime
- SortDistriNetworkID ::= ‘$x/upnp:channelID/@distriNetworkID’ ‘ascending’
- SortChannelID ::= ‘$x/upnp:channelID’ ‘ascending
- SortStartTime ::= ‘$x/upnp:scheduledStartTime’ ‘ascending’

When the FilterPropList notation is used, the specified properties will return like the Filter argument of CDS::Browse action. When “$x” is specified, all properties in each epgItem will return like the “*” specification in the argument. In the former case, the UPnP AV MediaServer that implements the EPG Server Device Option shall complement mandatory properties (i.e. @id, @parentID, etc) in the result document to keep the returned DIDL-Lite document valid. In the latter case, when the retuned item includes properties whose namespaces are not declared in the requested ConstructorSTag, the UPnP AV MediaServer that implements the EPG Server Device Option shall add appropriate namespace declarations in the returned DIDL-Lite start tag.

- ReturnClause ::= ‘return’ (‘<item>’ ‘{‘ FilterPropList ‘}’ ‘</item>’ | ‘$x’)
- FilterPropList ::= FilterProp (‘,’ FilterProp)*
- FilterProp ::= elem-spec | att-spec
- elem-spec ::= direct-elem-spec | nested-elem-spec
- direct-elem-spec ::= ‘$x/’element-name
- element-name ::= char*
- nested-elem-spec ::= ‘{‘ nested-elem-constBegin ‘{‘ nested-elem-val-spec ‘}’ nested-elem-constEnd ‘}’
- nested-elem-constBegin ::= elemBegin*
- elemBegin ::= ‘<’ element-name ‘>’
- nested-elem-constEnd ::= elemEnd*
- elemEnd ::= ‘</’ element-name ‘>’
- nested-elem-val-spec ::= ‘$x/’element-name’/’(element-name’/’)*’text()’
- att-spec ::= item-att-spec | property-att-specs
- item-att-spec ::= ‘$x/@’attribute-name
- attribute-name ::= char*
- property-att-specs ::= ‘{‘ nested-elem-constBegin ‘{‘ nested-elem-val-spec nested-elem-att-spec* ‘}’ nested-elem-constEnd ‘}’
- nested-elem-att-spec ::= ‘$x/’element-name’/’(element-name’/’)*’@’attribute-name
- ConstructorETag ::= ’</DIDL-Lite>’

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	GHIF3	N

NOTE: A UPnP AV MediaServer that implements the EPG Server Device Option does not need to support the full XQuery language. This guideline defines a subset of the XQuery language [46]. The goal of this subset is to allow for efficient implementations (on embedded platforms). This subset is referred to as level “DLNA_EPG”. In Annex K further explanation of this subset is provided.

7.4.6.7.6

[GUIDELINE] A UPnP AV MediaServer that implements the EPG Server Device Option may support the XQuery language subset defined by the following EBNF notation. This subset is referred to as the “DLNA_EPG_EXPANDED” XQuery level.

- QueryBody ::= NameSpaceDecl EnclosedExpression| prologConstructor
- NameSpaceDecl ::= DidlliteNSDecl DcNSDecl? UpnpNSDecl? OtherNSDecl*
- DidlliteNSDecl ::= ‘declare’ ‘default’ ‘namespace’DidlliteNSAttName;
- DcNSDecl ::= ‘declare’ ‘namespace’ DcNSAttName;
- UpnpNSDecl ::= ‘declare’ ‘namespace’ UpnpNSAttName;’
- OtherNSDecl ::= ‘declare’ ‘namespace’ OtherNSAttName;

Some of definitions are from 7.4.6.7.5.

- prologConstructor ::= ConstructorSTagEnclosedExpressionConstructorETag
- EnclosedExpression ::= ‘{‘ SingleMainExpr ‘}’
- SingleMainExpr ::= FlworExpr | (‘(‘ FlworExpr ‘)’ ‘[‘ ‘fn:position()’ ‘=’ Integer ‘to’ Integer ‘)’ ‘]’)
- FlworExpr ::= ForClause WhereClause? OrderByClause? ReturnClause
- ForClause ::= ‘for’ Variable ‘in’ ForExpr
- Variable ::= ‘$’ ‘VarName’
- VarName ::= QName
- QName ::= PrefixedName | UnprefixedName
- PrefixName ::= Prefix ‘:’ LocalPart
- UnprefixedName ::= LocalPart

Note: “Prefix” and “LocalPart” are from the XML specification. The detail definitions are defined in [69].

- ForExpr ::= (PathStart PathExpr) | ‘fn:distinct-values’ ‘(‘ PathStart PathExpr ‘)’
- PathStart ::= ‘DIDL-Lite//item’ | ‘DIDL-Lite//container’
- PathExpr ::= ((‘/’ Pname) | (‘//’ Pname) | Predicate)*
- Predictate ::= ‘[‘ Expr* ‘]’
- Pname ::= QName | (‘@’ UnprefixedName
- WhereClause ::= ‘where’ Expr
- Expr ::= Property | Comparison | (Expr ‘or’ Expr) | (Expr ‘and’ Expr) | (‘(’ Expr ‘)’) | Function
- Property ::= (Variable PathExpr) | PathExpr
- Comparison ::= Property CompOperator Literal
- CompOperator ::= ‘=’ | ‘!=’ | ‘<’ | ‘<=’ | ‘>’ | ‘>=’

“Function” is defined in [70].

- OrderByClause ::= ‘order by’ SortList
- SortList ::= (Property SortDirection?) | (‘,’ Property SortDirection?)*
- SortDirection ::= ‘Ascending’ | ‘Descending’
- ReturnClause ::= (‘return’ Variable PathExpr?) | (‘return’ Constructor)

- Constructor ::= (‘<’ UnprefixedName DirAttributeList? ‘/>’) | (‘<’ UnprefixedName DirAttributeList? ‘>’ DirElemContent ‘</’ UnprefixedName ‘>’ ReturnExpression?
- DirAttributeList ::= (UnprefixedName ‘=’ ‘"’ Literal ‘"’)+
- DirElemContent ::= ‘{‘ Expr (“,” Expr)* ‘}’
- ReturnExpression ::= ‘[‘ Expr* ‘]’
- Literal::=NumericLiteral | StringLiteral
- NumericLiteral::=IntegerLiteral | DecimalLiteral | DoubleLiteral
- IntegerLiteral::=Digits [195] DecimalLiteral::=("." Digits) | (Digits "." [0-9]*)
- DoubleLiteral::=(("." Digits) | (Digits ("." [0-9]*)?)) ("e" | "E") ("+" | "-")? Digits
- StringLiteral::=('"' (('"' '"') |[^"])* '"') | ("'" (("'" "'") | [^'])* "'")

[ATTRIBUTES]

O	R	DMS	M-DMS	n/a	[46] [69] [70]	PIUCS	N

NOTE: A UPnP AV MediaServer that implements the EPG Server Device Option does not need to support the full XQuery language. This requirement defines a subset of the XQuery language [46]. The goal of this subset is to allow for searching and retrieval of most typically needed EPG items. This subset is referred to as level “DLNA_EPG_ EXPANDED”. In Annex K further explanation of this subset is provided.

7.4.6.7.7

[GUIDELINE] A UPnP AV MediaServer that implements the EPG Server Device Option shall support the following properties in the ForClause, WhereClause, OrderByClause, and ReturnClause of the XQuery expression.

- dc:title
- upnp:channelName
- upnp:channelID
- upnp:scheduledStartTime

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	Q66BU	N

NOTE: The CDS:GetFreeFormQueryCapabilities action returns an XML-based capability list in the FFQCapabilities argument. This list defines which properties are allowed in the XQuery, hence it defines the minimum set of properties which can be used for searching, sorting, and filtering.

7.4.6.7.8

[GUIDELINE] A UPnP AV MediaServer that implements the “DLNA_EPG” level XQuery subset shall support the following additional properties in the ForClause, WhereClause and ReturnClause of the XQuery expression.

- dc:class
- upnp:channelID/@type
- upnp:scheduledEndTime
- upnp:scheduledDuration

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	74T34	N

NOTE: Sorting of these properties is optional.

7.4.6.7.9

[GUIDELINE] A UPnP AV MediaServer that implements the “DLNA_EPG_ EXPANDED” level XQuery subset shall support the following additional properties in the ForClause, WhereClause and ReturnClause of the XQuery expression.

- didl-lite:item/@id
- didl-lite:item/@parentid
- didl-lite:container/@id
- didl-lite:container/@parentid
- didl-lite:res/@protocolInfo
- didl-lite:res/@refID
- dc:creator
- upnp:class
- upnp:channelID/@type
- upnp:channelID/@distriNetworkName
- upnp:scheduledEndTime
- upnp:scheduledDuration
- upnp:longDescription

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	8DWDW	N

NOTE: Sorting of these properties is optional.

7.4.6.7.10

[GUIDELINE] A UPnP AV MediaServer that implements the EPG Server Device Option shall implement the CDS:GetFreeFormQueryCapabilities action as defined in [46].

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	85U35	N

7.4.6.7.11

[GUIDELINE] If the UPnP AV MediaServer supports a subset of XQuery the CDS:GetFreeFormQueryCapabilities action, and there are properties that are allowed anywhere in the XQuery except the “order-by” clause, it shall return a list defined by <searchOnlyPropertyList> This list shall at least contain the <propertyName> elements, defined in 7.4.6.7.5 unless that property is already listed in the <propertyList> (see [46]).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	OXGLO	N

NOTE: This list defines the properties that can be used for searching but not for sorting, and complements the list defined in [46]. When a UPnP AV MediaServer allows sorting on more properties than the minimal required set, e.g. sorting on upnp:class, these properties will be added to the <propertyList> and will not appear in the <searchOnlyPropertyList>.

7.4.6.7.12

[GUIDELINE] If a UPnP AVMediaServer allows for searching and sorting using foreign-metadata properties in an XQuery, then these properties shall be listed in the <propertyList> as defined in [46]. Each property shall be fully qualified as defined in [46].

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	TOUKX	N

7.4.6.7.13

[GUIDELINE] If a UPnP AVMediaServer allows for searching but not sorting using foreign-metadata properties in an XQuery, then these properties shall be listed in the <searchOnlyPropertyList> as defined in 7.4.6.7.11. Each property shall be fully qualified as defined in [46].

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	YFF39	N

7.4.6.7.14

[GUIDELINE] The <namespaceList> element in the FFQCapabilities argument returned by the CDS:GetFreeFormQueryCapabilities action shall list the namespaces used for the properties in the <propertyList> and the <searchOnlyPropertyList>. This includes the namespaces of foreign-metadata properties that are present in the <propertyList> or the <searchOnlyPropertyList>. It shall at least contain the following <namespaceName> elements:

- dc=http://purl.org/dc/elements/1.1/
- upnp=urn:schemas-upnp-org:metadata-1-0/upnp/
- didl-lite=urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	3MRW2	N

7.4.6.7.15

[GUIDELINE] If a UPnP AV MediaServer that implements the “DLNA_EPG” level XQuery subset cannot process a query because it does not conform to the subset defined in 7.4.6.7.5, it shall return the UPnP error 728 (Unsupported Query Request Instruction(s)).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	A42Y2	N

NOTE: UPnP AV MediaServer implementing more than the required “DLNA_EPG” level subset can execute a query it can understand. In this case no error is reported.

7.4.6.7.16

[GUIDELINE] If a UPnP AV MediaServer that implements the “DLNA_EPG_EXPANDED” level XQuery subset cannot process a query because it does not conform to the subset defined in 7.4.6.7.6, it shall return the UPnP error 726 (Unsupported Query Request Instruction(s)).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	I5TEA	N

NOTE: UPnP AV MediaServer implementing more than the required “DLNA_EPG” level subset can execute a query it can understand. In this case no error is reported.

7.4.6.7.17

[GUIDELINE] If the QueryRequest input argument of the CDS:FreeFormQuery action contains an XQuery that specifies a property that is not listed in neither the <propertyList> nor the

<searchOnlyPropertyList> as defined in 7.4.6.7.11, a UPnP AV MediaServer shall return the UPnP error 708 (Unsupported or invalid search criteria).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	DKXM4	N

7.4.6.7.18

[GUIDELINE] If the QueryRequest input argument of the CDS:FreeFormQuery action contains an XQuery that specifies in the “order by”-clause a property that is not listed in the <searchOnlyPropertyList> as defined in 7.4.6.7.11, a UPnP AV MediaServer shall return the UPnP error 708 (Unsupported or invalid search criteria).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[46]	WYN2C	N

7.4.6.7.19

[GUIDELINE] The CDSView input argument passed to the CDS:FreeFormQuery action shall be set to “0” indicating DIDL-Lite view.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[46]	XF9K8	N

7.4.6.8 MM/EPG Event Moderation

7.4.6.8.1

[GUIDELINE] A UPnP AV MediaServer that implements the EPG Server Device Option shall implement the ContainerUpdateIDs state variable.

[ATTRIBUTES]

M	L	DMS	M-DMS	n/a	[46]	KJNK4	N

NOTE: Control points can subscribe to CDS events. The ContainerUpdateIDs state variable signals changes to containers. This means that a subscribed control point will be notified in case of changes in the EPG container. The ContainerUpdateIDs state variable is a CSV list of ordered pairs, where each pair contains the @id of a container and its ContainerUpdateIDValue. The @id indicates the container in which a change occurred. If multiple changes occurred in a container since the last event was sent, there will be only one occurrence of the container’s @id and the ContainerUpdateIDValue will reflect the most recent change.

7.4.6.8.2

[GUIDELINE] When a large number of EPG items are changed, a UPnP AV MediaServer should not send ContainerUpdateIDs events before completion of the update.

[ATTRIBUTES]

S	L	DMS	M-DMS	n/a	[46]	5MGBH	N

NOTE: A large EPG update that sends interim update events can trigger UPnP AV control points to prematurely issue a search to obtain updated EPG information. To further reduce the chance of issuing searches while an EPG update is in progress, UPnP AV control points can wait a reasonable amount of time (to ensure a smooth user experience), before requesting new EPG data.

7.4.6.8.3

[GUIDELINE] A UPnP AV MediaServer that implements the EPG Server Device Option may implement the Track Changes Option as defined in [46].

[ATTRIBUTES]

O	L	DMS	M-DMS	n/a	[46]	ENS95	N

NOTE: A UPnP AV MediaServer can choose to event changes to individual objects. A UPnP AV control point can use such an event to take action when, for example, the broadcast time of an EPGItem was changed. When a large number of EPGItems are modified, added or deleted, the UPnP AV MediaServer can set the stUpdate attribute of individual objects to 1. A control point can make use of this information to ignore individual changes and only take action when it sees the stDone event.

7.5 Media Transport

7.5.1 General

This subclause of the DLNA Home Networked Device Interoperability Guidelines covers the requirements for the media transport layer. In the DLNA Interoperability Guidelines v1.0 there was a single transport protocol (HTTP) available for the transfer of media across a home network and all media transfers were assumed to be for the purpose of immediate playback. In addition, all media transfers under DLNA Interoperability Guidelines v1.0 occurred with a default, best effort, quality of service specification. With the increase in the System Usages in v1.5, the introduction of priority-based QoS, and the addition of another transport protocol (RTP), there exists a need to define different modes of media transfer and other protocol-agnostic requirements for the transport layer. Table 46 summarizes the types of media transfer now available. The DLNA transfer mode terms are consistent with those found in subclause 6.3 of [76].

Table 46 — DLNA Media Transfer Modes

Transfer Mode	Transfer Rate	Example Usages	Default DLNAQOS Level
Streaming Transfer	For Audio and AV streams the Content Source and Content Receiver maintain under Ideal Network Conditions an average transfer rate equal to or higher than the rate sufficient for real-time rendering.	Immediate rendering by the Content Receiver of content binaries with an inherent time base (e.g. Audio or AV media). Real time generated AV media transfer followed by store/record at the Content Receiver.	DLNAQOS_2
Interactive Transfer	The transfer rate is limited to the lesser of the maximum transfer rate of the Content Source and the maximum transfer rate of the Content Receiver without degrading any Streaming Transfer originating from the Content Source.	Immediate rendering by the Content Receiver of content binaries with no inherent time base (e.g. images or printer documents).	DLNAQOS_1
Background Transfer	The Content Source transfers the content binary at a rate determined by the Content Source, but no faster than the rate at which the Content Receiver can accept the content binary from the network.	Transfer and store of file-based media.	DLNAQOS_0 (Lowest Priority)

Table 47 summarizes the combinations of permitted DLNAQOS priorities and transfer modes for each media class. The relationship between the different columns is described here.

- Media Class: Indicates a media classe.
- Transfer Mode: Indicates a transfer mode.
- Combination Permitted: Indicates if the indicated Media Class and Transfer Mode values can be combined. The "Yes" and "No" values indicate if the combination is permitted. A "Default" value means that Content Sources are required to support the combination. The permissible combinations are described in the following guidelines:
    - 7.4.1.3.34 MM tm-s (Streaming Mode Transfer Flag)
    - 7.4.1.3.35 MM tm-i (Interactive Mode Transfer Flag)
    - 7.4.1.3.36 MM tm-b (Background Mode Transfer Flag)
- Permitted DLNAQOS_UP: Indicates the DLNAQOS_UP values that the Content Source is permitted to use when responding to transport requests. The guidelines do not require Content Sources to always respond with the highest DLNAQOS_UP value that is listed in this column. The following guidelines describe the permitted DLNAQOS_UP values for a given media class.
    - 7.5.4.2.3.2 in 7.5.4.2.3 MT Transfer Mode Support
    - 7.5.4.2.10 MT DLNAQOS Background Transfer
    - 7.5.4.2.11 MT DLNAQOS Interactive Transfer
    - 7.5.4.2.12 MT DLNAQOS Streaming Media

Table 47 — Permitted Combinations of DLNAQOS_UP and Transfer Mode Per Media Class

Media Class	Transfer Mode	Combination Permitted	Permitted DLNAQOS_UP
Image, Media Collection File, or XHTML Print Document	Streaming	No	n/a
	Interactive	Default	DLNAQOS_1, DLNAQOS_0
	Background	Yes	DLNAQOS_0
Audio	Streaming	Default	DLNAQOS_2, DLNAQOS_1, DLNAQOS_0
	Interactive	No	n/a
	Background	Yes	DLNAQOS_0
AV	Streaming	Default	DLNAQOS_2, DLNAQOS_1, DLNAQOS_0
	Interactive	No	n/a
	Background	Yes	DLNAQOS_0

A Streaming Transfer supports two media usages. The first case is where a content binary is being immediately rendered for a user and contains inherent timing that shall be met. The second case is where a content binary is being generated in real time at a fixed rate (such as a live broadcast stream), regardless of whether the item is being immediately rendered or stored for later use. In either of these cases, a delay in packet delivery can adversely impact the user's perception of the system. If the content binary contains inherent timing information (such as Audio or AV content) and is being immediately rendered, a delay in packet delivery can cause data to not be available on the Content Receiver at the time it needs to be played. This can lead to a dropout in the playback of the media. If a real time stream is being generated on the Content Source and the throughput across the network (which can be affected by the Content Receiver's use of flow control) is not equal to the data rate of the generated content, a buffer overrun can occur on the Content Source. This can lead to a loss of data in the content binary[1] .

An Interactive Transfer is used for the case where a content binary is being immediately rendered for a user but it does not have any inherent timing information, such as image media. In this case, a delay in packet delivery of a few milliseconds will not cause an adverse impact for rendering, but sufficiently long delays will adversely impact the user's perception of the system.

A Background Transfer is used for the case where the content binary is not being transferred for immediate rendering or where the user might be satisfied with a transfer executed at the lowest priority. It is typically reserved for the download or upload of content that is not being generated in real time by the Content Source. For example, this transfer mode would be used for downloading a content binary that has been stored in a file on the Content Source. The Content Source is free to internally prioritize the Background Transfer lower than other transfer modes.

The DLNA QoS levels are shown in the above tables as an indication that each of the transfer modes are handled at a different QoS level. Further discussion of QoS can be found in 7.2.

Each of the Transfer Modes is implemented differently for the transport chosen (HTTP vs. RTP). In addition, a transport may choose to not allow a particular Transfer Mode. For instance, RTP will not support Background Transfers. The Transfer Modes available to a particular content resource are specified in the Media Management subclause.

For most transfers, the Content Receiver issues a request for the content binary with a specific Transfer Mode and Media Transport based on the type of usage involved. An Upload content transfer is the exception to this rule. It is initiated by the Content Source.

However, the choice of Transfer Mode is made in the same way, based on the type of usage and the content binary to be transferred.

The definitions of the Transfer Modes introduced in this subclause apply only to a state of the network referred to as Ideal Network Conditions. Under Ideal Network Conditions no congestion from other communicating parties or from bandwidth restrictions on the network exists. Furthermore, under Ideal Network Conditions, Content Sources and Content Receivers have moved away from the startup conditions, and have reached steady-state transfers. From the perspective of measurements, Ideal Network Conditions can be approximated by having a single Content Source send data to a single Content Reveiver over a high-speed link with bandwidth equal to or exceeding the bandwidth required by the transmission.

This system provides a robust set of Transfer Modes that support streaming for immediate rendering or download of content for later consumption. It also provides a structure for fine grained QoS control. A DLNA Interoperability Guidelines v1.0 transfer most closely maps to a Streaming Transfer Mode in terms of functionality. However the Streaming Transfer Mode

1. If the content is being generated at an average fixed rate, such as the capture of audio or AV content from an external source, and the network has a period where the throughput it less than the rate of generation of the content, the Content Source will typically buffer the data for sending at a later time. However, if this period lasts long enough, the Content Source may exhaust its ability to buffer the captured content. At this point, some data samples will be lost. If the Content Receiver is rendering the stream immediately, the user will perceive the loss of the data as skipped content samples. If the Content Receiver is storing the stream for later use, such as in a download operation, the stored content will have missing samples. Content Sources and Receivers may have any amount of buffering to mitigate this situation and the network cannot be controlled to guarantee a bandwidth for the transfer. Any content that is transitory (i.e. not stored on disk) and may cause a buffer overrun on the Content Source due to network delays should always be sent as a Streaming Transfer.

allows for higher QoS priority than the default best effort of the DLNA Interoperability Guidelines v1.0.

7.5.2 Uniform Client Data Availability Model

The Uniform Client Data Availability Model (UCDAM) provides a mechanism for describing the data available for a content stream. It defines a content stream in mathematical terms, with special attention focused on the data range that can be transmitted by the Content Source. The model applies regardless of whether the content is stored, converted (e.g. transcoded, transrated, transscaled, etc.), or "live". Although the DLNA guidelines do not specify buffering implementations or requirements for either the Content Source or the Content Receiver, the uniform data availability model takes into account the diversity of buffering models that are available to implementers. These guidelines can then focus on normative high level behaviors that are common, regardless of details at the transport layer. Figure 32 graphically shows the UCDAM model.

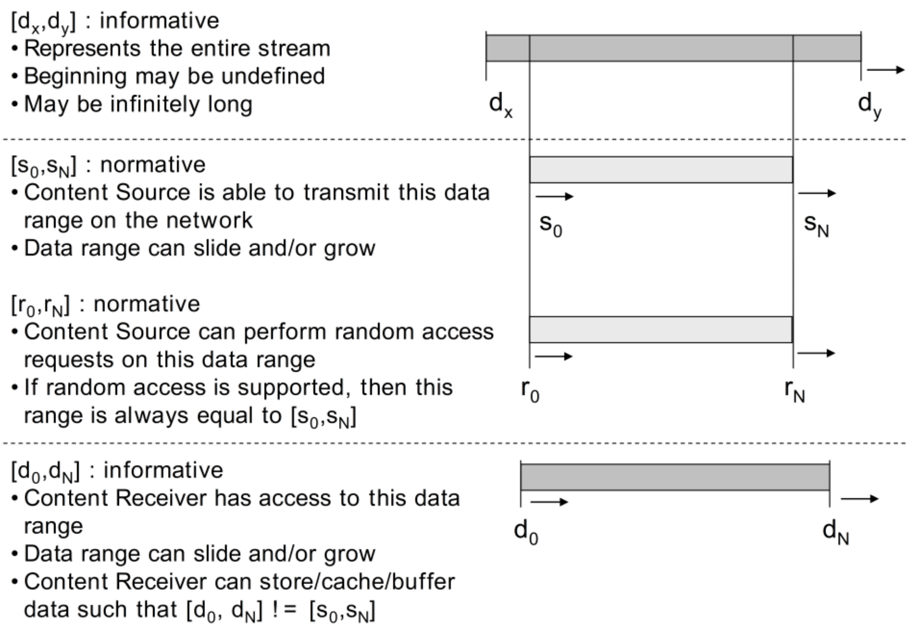


Figure 32 — UCDAM Summary

- The first aspect of the UCDAM is the definition of a *content stream*. A *content stream* is media with a beginning (d_X) and end (d_Y), where both values are defined by the Content Source. In some cases, a content stream never ends (d_Y increases with time). This data range is an assumed condition but is not referenced in the guidelines.
- The second aspect of the UCDAM is the *data range available from the Content Source* for transport to the Content Receiver. This range is denoted as [s_0, s_N]. For content stored within a file on the Content Source, s_0 could be equal to d_X and s_N could be equal to d_Y. For content captured by the Content Source from a live AV or audio feed, the range represents the amount of data buffered by the Content Source. This data range is normative and is referenced in the guidelines.
- The third aspect of the UCDAM is the *data range available to the Content Receiver*. The range of data available to the Content Receiver, defined inclusively as [d_0, d_N], is determined by two aspects: what the Content Source can transmit (i.e. some data range of

[s_0, s_N]) and, in addition, what might be buffered on the Content Receiver in a local manner[2]. This data range is an assumed condition but is not referenced in the guidelines.

- The fourth aspect of the UCDAM is the *Content Source's data range that supports random access requests*. The [r_0, r_N] data range represents the data range where random access operations are supported. If this capability is supported, then the [r_0, r_N] interval is always equal to the [s_0, s_N] data range. If the Content Source does not allow random access within the content stream, then a Content Receiver can only request content starting from s_0. This data range is normative and is referenced in the guidelines.
- The fifth aspect of the UCDAM is that the *data range available to the Content Receiver can change with time*. This is really a clarification of the third aspect. As time passes, the data range that the Content Source can provide to the Content Receiver can also change. There are three aspects that determine how the data range available to the Content Receiver will change.
  - Does the Content Source guarantee that s_0 is fixed or can s_0 increase with time? If s_0 is fixed, then the Content Source is characterized as operating under a fixed-s_0 model. Otherwise, the Content Source is characterized as operating under an increasing-s_0 model. An example of a possible fixed-s_0 model is content data read from a file on the Content Source. An increasing-s_0 model.can be used to represent content being captured from an incoming AV feed.
  - Does the Content Source guarantee that s_N is fixed or can s_N increase with time? If s_N is fixed, then the Content Source is characterized as operating under a fixed-s_N model. Otherwise, the Content Source is characterized as operating under an increasing-s_N model. The same examples as above can be used in this case. An example of a fixed-s_N model is content data read from a file on the Content Source. An increasing-s_N model.can be used to represent content being captured from an incoming AV feed.[3]
  - Does the Content Receiver *save data* [4] such that ($d_0 < s_0$)? (i.e. the Content Receiver has accessible data that the Content Source can no longer provide.)

Note that this model is very flexible in that it can easily represent all of the common types of media streams. For example, a media stream originating from a file can be represented as d_0=dx and d_N=d_Y - the entire extent of the content is available to the user. An unbuffered live stream can be represented as d_0=d_N, so that only the current moment in time is available. In addition to above simple examples, many more complex buffering systems can be represented by UCDAM. Please see Annex E for more details.

Keep in mind that most of the UCDAM model is conceptual and outside the scope of the DLNA Guidelines. However, the [s_0, s_N] and [r_0, r_N] data ranges describe what data can be requested in Media Transfer operations over the network, and as such are within the scope of these Guidelines. Therefore, those data ranges will be utilized in the following Guidelines where appropriate to define the range of content that can be transferred using DLNA defined Media Operations.

7.5.3 Media Operations

A Media Operation is the network level operation that supports a user interaction with a content binary. At a high level, they define the network operations that shall occur in order to

2. Neither the UCDAM model, nor the DLNA Media Transport guidelines specify how clients have access to data in a local manner as implementers may use a variety of memory-based and disk-based mechanisms to define the range of data that a Content Receiver can access without the Content Source having to transfer data.
3. The examples here are not intended to imply that the Content Source must use a particular model when representing file based or captured media, but should be regarded as illustrative of one possible implementation.
4. Neither the UCDAM model, nor the DLNA Media Transport guidelines require or specify how receiving endpoints save data.

support a Streaming Transfer mode data transfer. The core set of Media Operations that an endpoint can perform are defined in the following table.

Table 48 — DLNA Streaming Media Operation Definitions

Term	Definition
Play	Initiate a Streaming Transfer for playback of media. The transfer occurs at a rate that supports normal playback of the content binary. The transfer begins at s_0 and proceeds at a rate sufficient to support normal (1x) playback of the content binary. The operation completes after transfer of a fixed s_N value. Under the increasing-s_N model the transfer does not complete.
Stop	Terminate a Streaming Transfer.
Pause	Temporarily suspend a Streaming Transfer.
Pause-Release	A Pause media operation has suspended a Streaming Transfer, complete the Pause media operation and re-establish the flow of data over the network to support the Streaming Transfer.
Seek	ove the transfer point to a particular point in a stream in the range $[r_0, r_N]$, that represents the seek-able range. (The seek-able data range is the same random access data range, although the former term is used for Streaming transfers and the latter term applies to Streaming, Interactive, and Background transfers.) If a seek-able range exists, it shall equal $[s_0, s_N]$. The next set of transferred data will be from the indicated point in the content binary. DLNA defines two types of seek operations: • Byte-based seek: a seek operation where the transfer point is specified in units of bytes • • Time-based seek: a seek operation where the transfer point is specified in units of time
Fast Forward Scan	Perform data transfers that will support a positive play speed greater than 1x.
Slow Forward Scan	Perform data transfers that will support a positive play speed greater than 0 but less than 1x.
Fast Backward Scan	Perform data transfers that will support a negative play speed less than -1x.
Slow Backward Scan	Perform data transfers that will support a negative play speed less than 0 but greater than -1x.
Streaming Download	Initate a Streaming Transfer for the purpose of storing media for later playback. The transfer begins at s_0 and proceeds at a rate defined by the internal timing information of the content. The operation completes after transfer of a fixed s_N value. Under the increasing-s_N model the transfer does not complete until the Content Receiver terminates the transaction.

The listed media operations are defined in terms of how the media transfer occurs over the network for a given transport protocol in the guidelines below.

7.5.4 Media Transport Protocols

7.5.4.1 General

The following subclauses contain guidelines for the use of media transports in DLNA devices. The guidelines are organized into a table that covers requirements common among all media transports and then subclauses/tables that cover requirements for specific media transport protocols such as HTTP and others.

7.5.4.2 Media Transport Common Guidelines

7.5.4.2.1 MT Mandatory Transport Support

[GUIDELINE] Content Sources and Content Receivers shall support HTTP as the mandatory transport as specified in the following subclauses.

- 7.5.4.3.2 HTTP Transport: Common Requirements
- 7.5.4.3.3 HTTP Transport: Streaming Transfer Guidelines
- 7.5.4.3.4 HTTP Transport: Interactive Transfer Guidelines
- 7.5.4.3.5 HTTP Transport: Background Transfer Guidelines
- 7.5.4.3.6 HTTP Transport: POST Guidelines

[ATTRIBUTES]

M	A	DMP DMR DMPr DMS +PU+ +UP+ +DN+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	n/a	H9DD7

7.5.4.2.2 MT Optional Transport Support

[GUIDELINE] Content Sources and Content Receivers may support RTP as an optional transport as specified in subclause 7.5.4.4 RTP Media Transport.

[ATTRIBUTES]

O	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	n/a	FPWW5

7.5.4.2.3 MT Transfer Mode Support

7.5.4.2.3.1

[GUIDELINE] This guideline no longer applies.

[ATTRIBUTES]

						9DD7K

7.5.4.2.3.2

[GUIDELINE] A content binary shall be transferred using one of the Transfer Modes indicated in Table 49 for its Media Class.

The list of available Transfer Modes for a Media Class is as specified below. This may be further limited by the transport protocol chosen.

Table 49 — MT Media Class Transfer Modes

Media Class	Transfer Mode
XHTML Print Doc:	Background, Interactive
Media Collection Binary:	Background, Interactive
Image:	Background, Interactive
Audio-only:	Background, Streaming

Media Class	Transfer Mode
AV:	Background, Streaming

[ATTRIBUTES]

M	A	DMP DMR DMPr +UP+ +DN+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	n/a	PWW5H	

NOTE: This can be either a Content Receiver requesting a particular Transfer Mode for content that it will receive or it can be a Content Source specifying the Transfer Mode when performing an upload operation.

7.5.4.2.3.3

[GUIDELINE] A Content Source may constrain the transfer of audio-only or AV content binaries to Streaming Transfer only if it is not able to support the Background Transfer mode for that content binary and transport protocol.

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	n/a	RE5VM	

NOTE: For example, a Content Source might do this because it is capturing content from a live stream and could potentially overflow its buffer if the network transfer is handled below the nominal rate of the content.

7.5.4.2.3.4

[GUIDELINE] A Content Source shall indicate the Media Transfer Modes that are available for a content binary by setting the tm-s, tm-i, and tm-b flags in the 4th field of the protocolInfo.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	2V7PF	

NOTE: See guidelines 7.4.1.3.23.2, 7.4.1.3.34.1, 7.4.1.3.35.1, and 7.4.1.3.36 for more information.

7.5.4.2.3.5

[GUIDELINE] This guideline no longer applies.

[ATTRIBUTES]

						WW5HZ	

NOTE: See guidelines 7.4.1.3.23.2, 7.4.1.3.34.1, 7.4.1.3.35.1, and 7.4.1.3.36 for more information.

7.5.4.2.3.6

[GUIDELINE] An endpoint responding to the initiation of a media transfer shall generate an appropriate transport protocol error response if the requested Transfer Mode is not currently available for the given content binary.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	n/a	E5VMH	

NOTE: This can be either the Content Source responding to an incorrectly requested Transfer Mode by a Content Receiver or in the case of an upload operation it can be the Content Receiver responding to an incorrectly requested Transfer Mode by a Content Source. See guideline 7.5.4.3.2.33.4 for HTTP error codes used in various scenarios.

Guideline 7.5.4.2.3.2 defines the possible transfer mode given the media type and guideline 7.5.4.2.3.3 allows Content Sources to constrain the available transfer mode for a given item (for example if RTP is used as the transport protocol). This guideline requires an endpoint to generate an error if the request for the content is not of the allowed modes for the media item.

7.5.4.2.4 MT Transfer Mode Support for Device Classes

7.5.4.2.4.1

[GUIDELINE] A DMS, M-DMS, or +PU+ that supports the AV or Audio Media Classes shall support acting as a Content Source for Streaming Transfers as defined in this subclause

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	DD7KP	

7.5.4.2.4.2

[GUIDELINE] A DMS, M-DMS, or +PU+ that supports the Image Media Class shall support acting as a Content Source for Interactive Mode Transfers as defined in this subclause.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	V7PF7	

7.5.4.2.4.3

[GUIDELINE] A DMS or M-DMS that supports the download system usage shall support acting as a Content Source for Background Mode Transfers as defined in this subclause.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	n/a	7YPY8	

NOTE: The requirement is not that background will be used for the transfer. Rather, the guideline states that the Background transfer mode needs to be supported for one or more content binaries. See guideline 7.4.1.3.36 for more information about reporting Background transfer mode support for a content binary.

7.5.4.2.4.4

[GUIDELINE] A DMS or M-DMS that supports the upload AnyContainer or OCM: upload content operations shall support acting as a Content Receiver for Background Mode Transfers as part of the Content transfer process as defined in this subclause.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	n/a	W7YPY	

7.5.4.2.4.5

[GUIDELINE] A DMS or M-DMS that supports the upload AnyContainer or OCM: upload content operations should support acting as a Content Receiver for Streaming and Interactive Mode Transfers as part of the Content transfer process as defined in this subclause.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	n/a	HC83V	

NOTE: Allows the server to support the upload of content that cannot be sent via Background Transfer Mode such as live content captured from a tuner.

7.5.4.2.4.6

[GUIDELINE] A DMP, M-DMP, or DMR that supports the Audio or AV Media Classes shall support acting as a Content Receiver for Streaming Mode Transfers as defined in this subclause.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	n/a	n/a	BHC83	

7.5.4.2.4.7

[GUIDELINE] A DMP, M-DMP, or DMR that supports the Image Media Class shall support acting as a Content Receiver for Interactive Mode Transfers as defined in this subclause.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	n/a	n/a	78IWW	

7.5.4.2.4.8

[GUIDELINE] A DMPr shall support acting as a Content Receiver for Interactive Mode Transfers as defined in this subclause.

[ATTRIBUTES]

M	A	DMPr	n/a	n/a	n/a	XTQPS	

7.5.4.2.4.9

[GUIDELINE] An M-DMD, +DN+, or +DNSYNC+ shall support acting as a Content Receiver for Background Mode Transfers as defined in this subclause.

[ATTRIBUTES]

M	A	+DN+ +DNSYNC+	M-DMD	n/a	n/a	M93FM	

NOTE: The requestor isn't obligated to use background transfer if the server defines that only streaming of this content is supported.

7.5.4.2.4.10

[GUIDELINE] An M-DMD, +DN+, or +DNSYNC+ may support acting as a Content Receiver for Streaming Mode Transfers as defined in this subclause.

[ATTRIBUTES]

O	A	+DN+ +DNSYNC+	M-DMD	n/a	n/a	TQPSF	

NOTE: Supports the download of content that cannot be sent via Background Transfer Mode.

7.5.4.2.4.11

[GUIDELINE] An M-DMU, +UP+, or +UPSYNC+ shall support acting as a Content Source for Background Mode Transfers as defined in this subclause.

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	n/a	n/a	8IWW3	

7.5.4.2.4.12

[Guideline] A +PR1+ or +PR2+ shall support acting as a Content Source for Interactive Mode Transfers as defined in this subclause.

[Attributes]

M	A	+PR1+ +PR2+	n/a	n/a	n/a	AZ2ZF	

NOTE: XHTML documents and images are transferred under Interactive Mode by default.

7.5.4.2.5 MT Low Throughput Tolerance

[Guideline] Content Receivers shall tolerate scenarios where the Content Source is not able to sustain a particular transmission throughput.

Tolerate means that the Content Receiver is able to do one of the following things gracefully (i.e. without crashing or requiring the user to power-cycle or reset the device).

- continue receiving content data despite the low throughput, or
- terminate the transport layer connection.

[Attributes]

M	A	DMP DMR DMPr DMS +DN+ +DNSYNC+	M-DMS M-DMP M-DMD	MIU	n/a	93FM8	

NOTE: This guideline is mandatory because a home network does not always operate under Ideal Network Conditions (i.e. the transmission rate remains dependent on the network throughput between the server and client). Products that crash, require a reset, or a similar type of power-cycle operation as a result of low transmission throughput violate this guideline.

DLNA devices are permitted to have user interactions in meeting the tolerance portion of this requirement. For example, a DMP is permitted to report to ask the user if they want to stop rendering or a download because the throughput is extremely slow.

7.5.4.2.6 MT Requirements for Background Transfer

[Guideline] If Background Transfer is available for a given content binary, a downloading endpoint (an M-DMD, +DN+, or +DNSYNC+) shall use the Background Transfer Mode when performing a download operation.

[Attributes]

M	A	+DN+ +DNSYNC+	M-DMD	n/a	n/a	XAZ2Z	

NOTE: Examples of where this wouldn't be the case are the downloading of live media streams. On download, the server will mark them as only transportable with Streaming Transfer mode

7.5.4.2.7 MT Streaming Transfer Rate Assumptions

7.5.4.2.7.1

[Guideline] A Content Receiver operating in Streaming Transfer mode shall be able to receive content from the network at rates required to sustain Streaming Transfer for the profiles that the Content Receiver supports.

[Attributes]

M	A	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	n/a	5YE9G	

NOTE: A Content Receiver needs to be able to receive and consume content at a rate that will allow it to render the content in real-time. This guideline does not apply where the Content Source and Content Receiver have negotiated transfer characteristics within the transport protocol, such as negotiated buffer agreements within RTP. This guideline applies only in the absence of an existing agreement between the Content Receiver and the Content Source.

7.5.4.2.7.2

[GUIDELINE] A Content Source operating in Streaming Transfer mode shall be able to send content to the network at rates required to sustain Streaming Transfer for the content binary.

[ATTRIBUTES]

M	A	DMS +PU+ +UP+ +UPSYNC+	M-DMS M-DMU	n/a	n/a	3KP2E	

7.5.4.2.8 MT Interactive Transfer Rate Assumptions

7.5.4.2.8.1

[GUIDELINE] Content Sources and Content Receivers using an Interactive Transfer should tolerate all transmission throughputs.

In this case, tolerate means that Content Sources and Content Receivers do not terminate the transport connection simply because the throughput is too low, unless user intervention has caused it to happen.

[ATTRIBUTES]

S	A	DMS DMPr DMP DMR +PU+ +DN+ +PR1+ +PR2+ +DNSYNC+	M-DMS M-DMP M-DMD	MIU	n/a	Z5YE9	C

NOTE: This guideline covers the case of sending an image over the network for rendering; and it indicates that the transfer can occur at any rate depending on the network and server load. This guideline recommends that devices support a wide range of throughputs but the actual maximum and minimum depend on external factors like mechanical constraints in printers, etc.

7.5.4.2.8.2

[GUIDELINE] Content Sources and Content Receivers using an Interactive Transfer may affect the actual rate of data delivery using transport layer flow control, regardless of the content's internal timing information.

[ATTRIBUTES]

O	C	DMS DMP DMR +PU+ +DN+ +PR1+ +PR2+ +DNSYNC+	M-DMS M-DMP M-DMD	MIU	n/a	83KP2	A

7.5.4.2.8.3

[GUIDELINE] This guideline no longer applies.

[ATTRIBUTES]

						KP2EZ	

7.5.4.2.9 MT Background Transfer Rate Assumptions

7.5.4.2.9.1

[GUIDELINE] Content Sources and Content Receivers using a Background Transfer should tolerate all transmission throughputs.

In this case, tolerate means that Content Sources and Content Receivers do not terminate the transport connection simply because the throughput is too low, unless user intervention has caused it to happen.

[ATTRIBUTES]

S	A	DMS DMPr +PU+ +DN+ +UP+ +PR1+ +PR2+ +UPSYNC+ +DNSYNC+	M-DMS M-DMD M-DMU	MIU	n/a	E9GK5	C

NOTE: This guideline covers the case of downloading contentsourced from a file on the Content Source; and it indicates that the transfer can occur at any rate (e.g. higher or lower than the internal timing information of the content data for audio and A/V content). This guideline recommends that devices support a wide range of throughputs but the actual maximum and minimum depend on external factors like mechanical constraints in printers, etc.

7.5.4.2.9.2

[GUIDELINE] Content Sources and Content Receivers using a Background Transfer may affect the actual rate of data delivery using transport layer flow control, regardless of the content's internal timing information.

[ATTRIBUTES]

O	C	DMS DMPr +PU+ +DN+ +UP+ +PR1+ +PR-2+ +UPSYNC+ +DNSYNC+	M-DMS M-DMD M-DMU	MIU	n/a	3FM8E	A

7.5.4.2.10 MT DLNAQOS Background Transfer

7.5.4.2.10.1

[GUIDELINE] If DLNAQOS as defined in 7.2 is implemented, Background Transfer requests shall be tagged with DLNAQOS_0 in accordance with Table 11.

[ATTRIBUTES]

M	R	DMS DMPr +DN+ +DNSYNC+	M-DMS M-DMD	MIU	n/a	Z2ZFW	

7.5.4.2.10.2

[GUIDELINE] If DLNAQOS as defined in 7.2 is implemented, Background Transfers of content binaries shall be tagged with DLNAQOS_0 in accordance with Table 11.

[ATTRIBUTES]

M	R	DMS +UP+ +UPSYNC+	M-DMS M-DMU	n/a	n/a	2ZFW6	

7.5.4.2.10.3

[GUIDELINE] If DLNAQOS as defined in 7.2 is implemented by a Content Source and it receives a Background Transfer request for content that it cannot transfer at DLNAQOS_0, then it shall respond with an error within the transport used, at DLNAQOS_0, in accordance with Table 11.

For HTTP as a transport, see guideline 7.5.4.3.5.1.5 for the specific error.

[ATTRIBUTES]

M	R	DMS +PR1+ +PR2+	M-DMS	n/a	n/a	P2EZ3	

NOTE: For example, a Content Receiver that tries to use a Background Transfer to acquire a live stream might receive an error response from the Content Source because it cannot transmit a live stream at DLNAQOS_0.

Note that this guideline also applies to sourcing of XHTML-Print documents.

7.5.4.2.11 MT DLNAQOS Interactive Transfer

7.5.4.2.11.1

[GUIDELINE] If DLNAQOS as defined in 7.2 is implemented, Interactive transfer requests shall be tagged with DLNAQOS_1, or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3), in accordance with Table 11, independent of the transport used. Note that this guideline applies only when the transfer request is not marked as a Background Transfer.

[ATTRIBUTES]

M	R	DMPr DMP DMR	M-DMP	MIU	n/a	YE9GK	

NOTE: This transfer is part of an interactive experience and therefore the default is a higher QoS level than a Background Transfer.

7.5.4.2.11.2

[GUIDELINE] If DLNAQOS as defined in 7.2 is implemented, Interactive Transfers of content binaries shall be tagged with DLNAQOS_1, or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3), in accordance with Table 11, independent of the transport used.

[ATTRIBUTES]

M	R	DMS +PU+ +PR2+ +PR1+	M-DMS	n/a	n/a	FM8EP	

7.5.4.2.12 MT DLNAQOS Streaming Media

7.5.4.2.12.1

[GUIDELINE] If DLNAQOS as defined in 7.2 is implemented, Streaming Transfer requests shall use DLNAQOS_2, or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3), in accordance with Table 11.

[ATTRIBUTES]

M	R	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	n/a	C83VQ	

NOTE: For example, a Client Endpoint issues an HTTP GET request for AV content with DLNAQOS_2. The Content Source will then respond to the request with media that is tagged with DLNAQOS_2. The Content Source response (transfer of the actual media) will be tagged with DLNAQOS_2. Subsequent TCP ACK messages will use the existing TCP connection and therefore be tagged with DLNAQOS_2.

For the RTP transport, this priority is applicable to both audio and video streams encompassing TS, PS, and ES formats.

See 7.2.4.2.2 for considerations around the TCP connection establishment phase.

7.5.4.2.12.2

[GUIDELINE] If DLNAQOS as defined in 7.2 is implemented, Streaming Transfer of content binaries shall use DLNAQOS_2, or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3), in accordance with Table 11.

[ATTRIBUTES]

M	R	DMS +PU+	M-DMS	n/a	n/a	WW3DT

7.5.4.2.13 MT Normative Syntax for npt-time

[GUIDELINE] The syntax of the npt-time token shall be as follows:

- npt time = npt sec | npt hhmmss
- npt sec = 1*DIGIT ["." 1*3DIGIT]
- npt hhmmss = npt hh ":" npt mm ":" npt ss ["." 1*3DIGIT]
- npt hh = 1*DIGIT ; any positive number
- npt mm = 1*2DIGIT ; 0-59
- npt ss = 1*2DIGIT ; 0-59

[ATTRIBUTES]

M	L	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[27]	83VQW

NOTE: This guideline provides a consistent syntax for NPT time positions for both DLNA's extensions to HTTP and RTP Media Transport.

7.5.4.2.14 MT Normative Random Access Data Availability Models

7.5.4.2.14.1

[GUIDELINE] If a Content Source supports random access requests on a content binary, then the Content Source shall use only one of the following random access data availability models.

- “Full Random Access Data Availability" model, as defined in 7.5.4.2.15.
- "Limited Random Access Data Availability" model, as defined in 7.5.4.2.16.

These random access data availability models shall be used in a mutually exclusive manner.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+	M-DMS	MIU	n/a	YPY8R

NOTE: Previous versions of the DLNA guidelines do not formally define either random access data model, but the "Full Random Access Data Availability" model has been defined to match the assumptions used in previous versions of the DLNA guidelines.

Other guidelines explain how to detect which random access data availability model is being used. Specifically, the op-param is tied solely to the "Full Random Access Data Availability" model and the lop-npt/lop-bytes flags are tied solely to the "Limited Random Access Data Availability". For more information, see the following guidelines.

- 7.4.1.3.18 MM op-param (Operations Parameter - Common Guidelines)
- 7.4.1.3.28 MM lop-npt, lop-bytes and lop-cleartextbytes (Limited Operations Flags): Common

Note that the "Full Random Access Data Availability" model is the only model that can be used for Content Sources when serving image content or XHTML print documents. This limitation is inherent to the nature of such content, which neither have changing [s_0, s_N] data bounaries nor have any sender-pacing requirements.

7.5.4.2.14.2

[GUIDELINE] The UCDAM data range of [s_0, s_N] shall represent the entire data range that the Content Source can serve to other network endpoints.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+	M-DMS	MIU	n/a	7PF7U	

NOTE: Although not testable by itself, these guidelines repeat normative portions Figure 32. Other DLNA guidelines can normatively refer to the [s_0, s_N] data range.

With regards to how the s_0 and s_N boundaries change, other DLNA guidelines explain how to represent these abstract data boundaries in terms of zero-based byte indices or npt playback positions.

Generally, the s_0 and s_N data boundaries increase with time, although in some scenarios the values can reset, as is sometimes necessary when the integer value rolls over. Other guidelines specify the details on how these data boundaries can change.

7.5.4.2.14.3

[GUIDELINE] The [s_0, s_N] may change with time. Specifically, the following can happen.

- The s_0 data boundary may change with time.
- The s_N data boundary may change with time.

How these data boundaries change with time is undefined by the guidelines because these data boundaries are abstract. In some cases, other DLNA guidelines will impose restrictions that require a data boundary to remain fixed.

[ATTRIBUTES]

O	A	DMS +PU+ +PR1+	M-DMS	MIU	n/a	PY8RW	

7.5.4.2.14.4

[GUIDELINE] If a Content Source supports random access requests on content data, then the following rules shall apply.

- The UCDAM data range of [r_0, r_N] shall represent the data range where random access operations are permitted.
- The [r_0, r_N] data range shall be equal to the [s_0, s_N] data range.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+	M-DMS	MIU	n/a	PF7UZ	

NOTE: If random access requests are supported, then they need to be supported for the entire range that the Content Source can access.

This guideline is consistent with the following introductory material.

- Figure 32 — UCDAM Summary
- The prerequisites used for the Seek Media Operation definition, in Table 46 — DLNA Media Transfer Modes

7.5.4.2.15 MT "Full Random Access Data Availability" Model

7.5.4.2.15.1

[GUIDELINE] If a Content Source uses the "Full Random Access Data Availability" model, then following rules shall apply.

- The entire content binary shall be defined as the [s_0, s_N] data range.
- The s_0 data boundary shall map to a fixed and non-changing beginning. This requirement is a restriction of 7.5.4.2.14.3.
- The data range of [s_0, s_N] shall occupy an npt range of [0, npt-last-time] and a byte range of [0, last-byte-pos], where npt-last-time is in units of npt and last-byte-pos is in units of bytes.
- The content binary's zero position (i.e. npt-time=0 and byte-pos=0) shall map to the UCDAM's data position of s_0.
- The last-byte-pos and npt-last-time shall map to the UCDAM's s_N data position and the s_N data boundary shall map to the end of the available content data. (This requirement works in conjunction with 7.5.4.2.14.3.)
- The [r_0, r_N] and [s_0, s_N] data ranges shall have the same equality.
- Responses to random access requests on the [r_0, r_N] data range shall be timely under Ideal Network Conditions.

Timely means that the Content Source (under Ideal Network Conditions) is able to begin responding with the requested content data within 27 seconds of receiving the request.

Note that the npt-last-time, last-byte-pos, and byte-pos tokens for this guideline are relative to the complete content binary that is currently available, rather than being relative to the content data returned in response.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+	M-DMS	MIU	n/a	D7KPW	

NOTE: This guideline defines the behavioral model for the "Full Random Access Data Availability" model.

For guidelines on the data range for HTTP under "Full Random Access Data Availability", see the following guidelines.

- 7.5.4.3.2.18 MT HTTP Common Random Access Data Availability Requirements
- 7.5.4.3.2.19 MT HTTP Data Range of "Full Random Access Data Availability"

The “relative to the complete content binary” phrase means that the tokens apply in the context of the whole content binary. It is incorrect to interpret these tokens as they are used in actual response data. For example, if last-byte-pos=100 then it is correct to conclude that the complete content binary currently has 101 bytes. It is not necessarily true that a response with the Content-Range header's last-byte-pos=50 means that the complete content binary has 51 bytes because the last-byte-pos token in this context simply means that the last byte in the entity body is the 51st byte of the complete content binary.

7.5.4.2.15.2

[GUIDELINE] The values for npt-last-time and last-byte-pos (as specifically used in 7.5.4.2.15.1) may increase with time.

[ATTRIBUTES]

O	A	DMS, +PU+	M-DMS	MIU	n/a	5VMHZ	

NOTE: The concept of entire content range is relative to the current moment in time, in the context for the op-param. This means the s_N position can increase with time, causing the duration of the content binary to also increase (although the beginning has to remain fixed).

This model can apply when streaming content that is being recorded to local storage. The absolute beginning (s_0) never changes and as time passes, the end (s_N) increases.

7.5.4.2.16 MT "Limited Random Access Data Availability" Model

7.5.4.2.16.1

[GUIDELINE] If a Content Source uses the "Limited Random Access Data Availability" model, then only one of the two modes of operation shall be used.

- Mode=0
- Mode=1

Furthermore, the following rules shall be true for both Mode=0 and Mode=1.

- The [r_0, r_N] and [s_0, s_N] data ranges shall have the same equality.
- The [r_0, r_N] data range shall be the limited data range.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	n/a	7KPWX

NOTE: This guideline explains the behavior of the s_0 data boundary when it is used with the "Limited Random Access Data Availability" model.

For guidelines on the data range for HTTP under "Limited Random Access Data Availability", see the following guidelines.

- 7.5.4.3.2.18 MT HTTP Common Random Access Data Availability Requirements
- 7.5.4.3.2.20 MT HTTP: Data Range of "Limited Random Access Data Availability"

Note that the limited data range is the data range that supports seek media operations, as clarified in the other rows of this guideline.

7.5.4.2.16.2

[GUIDELINE] If a Content Source uses the "Limited Random Access Data Availability" model under Mode=0, then the following shall be true.

- The s_0 data boundary shall map to a beginning that shall change with time.
- The data range of [s_0, s_N] shall map to the npt range of [npt-start-time, npt-last-time] and the byte range of [first-byte-pos, last-byte-pos], where npt-start-time and npt-last-time are in units of npt and first-byte-pos and last-byte-pos are in units of bytes.
- There exists a "live position" that shall be equal to the s_N data boundary.
- If the s_N data boundary is changing with time, then the "live position" shall shift forward in real-time.
- Responses to random access requests on the [r_0, r_N] data range shall be timely under Ideal Network Conditions.
- If the Content Source receives a transport layer request that is not a random access request (e.g. HTTP request that omits Range and TimeSeekRange.dlna.org) then the Content Source shall respond with content data from the "live position". (See 7.5.4.3.2.20.9 for an example of how this guideline applies specifically to HTTP.)

Timely means that the Content Source (under Ideal Network Conditions) is able to begin responding with the requested content data within 27 seconds of receiving the request.

Real-time is the data rate necessary for immediate rendering.

Note that the npt-start-time, npt-last-time, first-byte-pos, last-byte-pos, and byte-pos tokens for this guideline are relative to the complete content binary that is currently available, rather than being relative to the content data returned in response.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	n/a	W5HZV	

NOTE: This guideline defines Mode=0 behaviors for the "Limited Random Access Data Availability" model. This mode of operation is generally useful for live content streams that use a fixed data buffer that map to the [s_0, s_N] and [r_0, r_N] data ranges. Live television broadcast streams are ideal candidates for this data availability model.

The reason why [r_0, r_N] is a limited data range in this context is that the values for npt-start-time, npt-last-time, first-byte-pos, and last-byte-pos change over time. For example, the value of first-byte-pos changes

- At time-0, the first-byte-pos is 1024. Random access requests that attempt to access before byte position 1024 will not work.
- At time-60, the first-byte-pos becomes 14749767106. Random access requests that attempt to access before byte position 14749767106 will not work, even though byte position 1024 was valid at time-60.

Please see the comment in 7.5.4.2.15.1 of 7.5.4.2.15 MT "Full Random Access Data Availability" Model for help with interpreting the "relative to the complete content binary" phrase.

7.5.4.2.16.3

[GUIDELINE] If using "Limited Random Access Data Availability" Mode=0, then a Content Source shall use increasing values for npt-start-time and first-byte-pos when reporting the available random access data range, unless one of the following conditions is true.

- 180 minutes has elapsed since the last transport layer request for the content binary
- The value of last-byte-pos or npt-last-time causes a rollover because the maximum permitted value defined for the data type has been exceeded.

This guideline only applies when "Limited Random Access Data Availability" applies to the scenario.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	n/a	KPWXV	

NOTE: As time passes, the data range that is accessible can also change with time because the server's s_0 position can change.

Furthermore the s_N position can increase with time, usually causing the duration of the content binary to either temporarily increase when s_0 is temporarily non-changing or remain relatively constant over time when s_0 slides with s_N.

Given the nature of infinite streams (e.g. live content), Content Sources are permitted to use a lesser values for first-byte-pos and npt-start-time.

The following example is an example timeline that exhibits this behavior.

- At time-0, the first-byte-pos is 0.
- Content requests occur for 180 minutes.
- At time-180, the first-byte-pos is 1474976710655.
- No content requests for 180 minutes.
- At time-360, the first-byte-pos is 0.
- Content requests occur for 360 minutes.
- At time-720, the last-byte-pos exceeds 281474976710655, so the server changes first-byte-pos to 0.

Also that a Content Receiver endpoint cannot seek to a position which it remembers if it does not request the content for over 180 minutes.

7.5.4.2.16.4

[GUIDELINE] If a Content Source uses the "Limited Random Access Data Availability" model under Mode=1, then the following shall be true.

- The s_0 data boundary shall map to a fixed and non-changing beginning. This requirement is a restriction of 7.5.4.2.14.3.
- The s_0 data position shall be the static and absolute beginning for the content. (i.e. The s_0 position is the beginning of the content that does not change with time.)
- The data range of [s_0, s_N] shall map to the npt range of [0, npt-last-time] and the byte range of [0, last-byte-pos], where npt-last time is in units of npt and last-byte-pos is in units of bytes.
- The content binary's zero position (i.e. npt-time=0 and byte-pos=0) shall map to the UCDAM's data position of s_0.
- The last-byte-pos and npt-last-time shall map to the UCDAM's s_N data position and the s_N data boundary shall map to the end of the available content data. (This requirement works in conjunction with 7.5.4.2.14.3.)
- Random access operations on [r_0, r_N], where units are in npt, shall be timely for the entire range of [r_0, r_N].
- Random access operations on [r_0, r_N], where units are in bytes, shall be timely only for a limited subset of [r_0, r_N]. Random access operations outside this subset are guaranteed to be satisfied but timeliness is not guaranteed.
- If the Content Source receives a transport layer request that is not a random access request (e.g. HTTP request that omits Range and TimeSeekRange.dlna.org) then the Content Source shall respond with content data from the beginning (i.e. npt=0 or byte-pos=0).

Timely means that the Content Source (under Ideal Network Conditions) is able to begin responding with the requested content data within 27 seconds of receiving the request.

Note that the npt-last-time, npt-time, last-byte-pos, and byte-pos tokens for this guideline are relative to the complete content binary that is currently available, rather than being relative to the content data returned in response.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	MIU	n/a	VMHZL

NOTE: This guideline defines Mode=1 behaviors for the "Limited Random Access Data Availability" model. Mode=1 is most useful for converted content, where Content Sources have access to a fixed beginning for the content but can lack the computational power to respond in a timely manner to byte-based random access positions. In such cases, the Content Source's indicated values for bytes-range represent the range where the server can provide a fast response, which is often the portion of converted content that is available at the current moment. Content Receivers are permitted to request data outside those ranges (assuming the requested data falls within the [s_0, s_N] data range, but the server might have a significant delay before responding.

Note that live broadcast streams are generally not ideal candidates for using this data availability model. Live broadcast streams will likely favor a sliding s_0 data boundary. For implementations that can support live broadcast streams where the s_0 data boundary is fixed (e.g. by buffering/recording the data to a local hard disk), the "Full Random Access Data Availability" model is more appropriate when the buffered content is served without having to go through a content transformation.

Please see the comment in 7.5.4.2.15.1 of 7.5.4.2.15 MT "Full Random Access Data Availability" Model for help with interpreting the “relative to the complete content binary” phrase.

7.5.4.3 HTTP Transport

7.5.4.3.1 General

There are many possible transport protocols that can be used for the transfer of content. The baseline mandatory media transport protocol for DLNA devices is HTTP. For HTTP the following terms are used:

- HTTP Client Endpoint - the DLNA entity that issues the HTTP GET or POST request
- HTTP Server Endpoint - the DLNA entity that receives the HTTP GET or POST request and issues an HTTP response (possibly including data).
- Streaming HTTP (Client/Server) Endpoint - An HTTP Client or Server Endpoint that processes Streaming Transfers.
- Target Response: When a client makes a GET request to obtain a certain resource from the server, the server normally responds with a message that includes a representation of the resource as its entity body. This type of response is called here a Target Response to differentiate it from other equally valid responses that do not involve transferring the requested resource (e.g., redirections, authorization requests, error messages, etc). Similarly, for HEAD requests, a Target Response is the same response that servers would form to satisfy the matching GET request, but without carrying the resource representation as its entity body.

The generic term "HTTP endpoint" is used to represent either an HTTP Client Endpoint or an HTTP Server Endpoint.

Also note that the guidelines specified in this subclause apply to DLNA content transactions between DLNA Device Classes or Device Capabilities. These guidelines do not specify behavior for non-DLNA devices. A DLNA Device Class or Device Capability may be implemented by software running on a more general-purpose device/platform. For example, the HTTP server of a DMS may be used to serve DLNA and non-DLNA content. These guidelines apply only when the DLNA content is being served to a DLNA device.

This subclause is organized into the following subclauses.

- 7.5.4.3.2 HTTP Transport: Common Requirements: This subclause contains guidelines that are common to all HTTP transfers. These guidelines are independent of the Transfer Mode.
- 7.5.4.3.3 HTTP Transport: Streaming Transfer Guidelines: This subclause contains guidelines that are specific to the use of HTTP for Streaming Transfers.
- 7.5.4.3.4 HTTP Transport: Interactive Transfer Guidelines: This subclause contains guidelines that are specific to the use of HTTP for Interactive Transfers.
- 7.5.4.3.5 HTTP Transport: Background Transfer Guidelines: This subclause contains guidelines that are specific to the use of HTTP for Background Transfers.
- 7.5.4.3.6 HTTP Transport: POST Guidelines: This subclause contains guidelines that are specific to HTTP POST transactions. HTTP POST transactions always work in conjunction with all other applicable HTTP guidelines.

7.5.4.3.2 HTTP Transport: Common Requirements

7.5.4.3.2.1 MT Baseline Transport: HTTP

7.5.4.3.2.1.1

[GUIDELINE] A DLNA device shall implement HTTP as the mandatory media transport with constraints and extensions defined in subsequent entries of this subclause.

Guidelines 7.5.4.3.2.7 and 7.5.4.3.2.26 define the HTTP version expectations for HTTP servers and clients.

[ATTRIBUTES]

M	L	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+. +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33]	5HZVT

NOTE: DLNA specifies HTTP as the baseline media transport.

7.5.4.3.2.1.2

[GUIDELINE] DLNA devices shall follow the syntax rules for HTTP headers defined in [33] unless DLNA defines syntax for the HTTP header value.

[ATTRIBUTES]

M	R	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+. +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33]	MHZLQ

NOTE: The BNF rules used in [33] is slightly different from that in DLNA. For example, the default treatment of literals in [33] is case-insensitive. Furthermore, [33] uses implied LWS between tokens and separator.

7.5.4.3.2.1.3

[GUIDELINE] If the DLNA guidelines define a BNF syntax for an HTTP header, then DLNA devices shall not include white spaces in the header-value of HTTP headers unless SP and LWS are explicitly specified in the syntax (BNF) definitions.

If the DLNA guidelines do not define a BNF syntax for an HTTP header, then the header shall conform to the message-header syntax in subclause 4.2 of [33], regardless of whether the HTTP header is defined in [33] or if the HTTP header is vendor-defined. Note that the syntax for field-value permits LWS to separate tokens and other data in the field-value.

Implied LWS between the HTTP header-name and the HTTP header-value are allowed as specified in [33], regardless of whether the DLNA guidelines specify a BNF syntax for the HTTP header.

The following cases are allowed examples:

- Range: bytes=1539686400-
- Content-Range:bytes 21010-47021/47022
- TimeSeekRange.dlna.org : npt=00:05:35.3-00

The following examples are not allowed:

- Range: bytes = 1539686400-
- Content-Range:bytes 21010-47021 / 47022
- TimeSeekRange.dlna.org : npt = 00 : 05 : 35.3-00

[ATTRIBUTES]

M	L	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+. +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33]	HZVTL

NOTE: [33] allows, including white spaces, between any two adjacent words (token or quoted-string), and between adjacent words and separators, e.g. "=", SP, "/", ":" (See "implied *LWS" in the subclause 2.1 of [33]), but this guideline restricts "implied *LWS" to simplify white space rules for HTTP headers.

White spaces can still be included between header-name and header-value. The header-name and header-value are defined in the subclause 4.2 of [33].

This guideline applies to HTTP headers defined in [33] and other HTTP headers, e.g, DLNA and vendor defined headers, with the DLNA guidelines. At least one LWS will be inserted between tokens in case of the "* rule" syntax (e.g. USER-AGENT and SERVER headers).

7.5.4.3.2.2 MT HTTP Graceful Recovery

[GUIDELINE] HTTP Client or Server Endpoints should not require a hardware reset or a power cycle to return to normal operating conditions after encountering improperly terminated HTTP connections.

[ATTRIBUTES]

S	A	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	n/a	ZVTLZ

NOTE: This guideline specifies that a media endpoint needs to be able to handle scenarios where an HTTP connection is not properly terminated. Network conditions and/or HTTP Server Endpoint behavior can cause this scenario to occur. Although a full definition for graceful recovery is not provided, a baseline expectation is that users will not need to reset or power cycle the device simply because a content transfer was interrupted.

7.5.4.3.2.3 MT HTTP DLNA URI Usage

[GUIDELINE] HTTP Client Endpoints that issue HTTP requests on DLNA URIs (such as those obtained from a UPnP AV ContentDirectory service implementation) may assume that the URI value is properly URI escaped. No additional URI escaping logic is required of a client endpoint.

[ATTRIBUTES]

O	C	DMP DMR DMPr +PU+ +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[29] [33] [49]	8RWS4

NOTE: This guideline permits an endpoint to use URI values (obtained from a DLNA endpoint) without having to implement logic for escaping the URI. This guideline is a clarification of 7.4.1.3.10, which states that DMS devices will advertise URI values that are URI escaped. Although an endpoint can implement additional logic for validating a URI, such logic is useful only for interoperation with non DLNA devices.

This guideline applies generally, including HTTP GET, HEAD, and POST requests.

7.5.4.3.2.4 MT Valid HTTP Response

7.5.4.3.2.4.1

[GUIDELINE] If an HTTP Server Endpoint receives an HTTP request, then the endpoint shall send a valid HTTP response provided it has sufficient platform resources (network sockets, stored file in readable state, available tuner hardware, etc.) for sending the response.

[ATTRIBUTES]

M	R	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33] [29]	HZLQD

NOTE: This guideline essentially obligates an endpoint to send an HTTP response to an HTTP request.

also, the HTTP specification already obligates a server to return a valid HTTP response for each received HTTP request. Valid HTTP responses include among others: content byte data responses, requests for authorization, HTTP error responses, etc.

7.5.4.3.2.4.2

[GUIDELINE] If an HTTP Server Endpoint cannot respond to an HTTP request by sending or receiving content byte data due to the server capacity, network capacity, or current state of the device (such as a tuner locked in a recording state), then the HTTP server should respond with an HTTP error response code of 503 (Service Unavailable).

[ATTRIBUTES]

S	R	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	F7UZS

NOTE: This guideline covers the case where the endpoint has the resources to send an error response but lacks the resources to send or accept content data. Sending an HTTP error is better than denying the TCP connection request because it explicitly tells the requesting endpoint that content cannot be handled at this moment. HTTP servers will respond with other HTTP error codes when responding to other error scenarios, as indicated in the HTTP specification.

7.5.4.3.2.4.3

[GUIDELINE] If an HTTP Server Endpoint cannot respond to an HTTP request by sending or receiving content byte data due to the device's lack of available network sockets, then the endpoint may refuse to create new TCP connections for answering content requests.

[ATTRIBUTES]

O	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[14] [33]	PWXV6

NOTE: This guideline permits an HTTP server to refuse the creation of new TCP connections when it lacks the resources for transporting content. Although this behavior is allowed by standard convention, endpoints can continue to retry the creation of a TCP connection. Therefore, whenever the situation is both appropriate and possible, HTTP servers are encouraged to respond with an HTTP 503 error.

7.5.4.3.2.4.4

[GUIDELINE] HTTP Client Endpoints that issue requests (e.g. for playback, download operation, or any normative system usage) shall be capable of performing such operations even if the HTTP Server Endpoint accepts only a single open HTTP connection at any given time.

HTTP Client Endpoints that claim support for a media operation for a particular content binary shall be capable of perfoming such operations even if the HTTP Server Endpoint accepts only a single open HTTP connection an any given time.

[ATTRIBUTES]

M	C	DMP DMR DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	[33]	3DTU5

NOTE: Some HTTP Server Endpoints can accept multiple simultaneous HTTP connections, but others can accept only one. This guideline ensures that HTTP Client Endpoints provide reliable services even with the most constrained case (only one HTTP connection available).

For example, the following procedures in HTTP Client Endpoints will not work with HTTP Server Endpoints that support a single HTTP connection:

- Obtaining an IFO file in parallel to a content transfer connection,
- Playback transitions between normal speed playback and trick mode playback with multiple HTTP sessions that overlap in time, and
- Tuner channel changes where one HTTP connection is used for the current channel and another time-overlapping HTTP connection is used for the new channel selection.

7.5.4.3.2.4.5

[GUIDELINE] HTTP Server Endpoints should support more than one simultaneous HTTP media transport connection.

[ATTRIBUTES]

S	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	7UZS4

NOTE: It is a good practice for an HTTP Server Endpoint to support multiple HTTP Client Endpoints simultaneously.

7.5.4.3.2.4.6

[GUIDELINE] If an HTTP Server Endpoint has not completed the transmission of an HTTP response and the HTTP Client Endpoint wants to stop the current data flow to issue a new request, then the HTTP Client Endpoint should close the existing TCP connection and then create a new TCP connection for the new HTTP request.

[ATTRIBUTES]

S	C	DMP DMR DMPr +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33]	Y8RWS

NOTE: Since clients cannot assume that endpoints can support multiple HTTP connections, this guideline recommends that clients use one TCP connection. Implementers of HTTP servers and clients should also consider this guideline in conjunction with guidelines 7.5.4.3.3.12.1 and 7.5.4.3.3.12.2, which deal with scan operation playback (a.k.a. trick-modes) for streaming transfers.

7.5.4.3.2.4.7

[GUIDELINE] An HTTP Server Endpoint shall begin sending a response message to an HTTP requests within 27 seconds of receiving the request. Valid response messages shall be either the response with content byte data, or appropriate error messages.

[ATTRIBUTES]

M	L	DMS +PU+ +PR1+ +PR2+	MDMS	n/a	[33]	VQWR3

NOTE: This guideline defines the maximum response time for an HTTP Server Endpoint for a request for content.

In conjunction with 7.5.4.3.2.4.9, these guidelines ensure HTTP transactions for media transport.

Also, that the time-out value is for the worst case. HTTP Server Endpoints are urged to respond to a request as soon as possible.

7.5.4.3.2.4.8

[GUIDELINE] An HTTP Client Endpoint shall wait at least 30 seconds before closing the TCP connection if it has not received any response from the HTTP Server Endpoint to an HTTP GET request for content.

[Attributes]

M	L	DMP DMR DMPr +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33]	3VQWR

NOTE: This guideline is not subject to scenarios involving user cancellation. A connection can be cancelled through user intervention at any time.

This does not imply that the entire content binary will be received within the 30 seconds, only that it has started.

7.5.4.3.2.4.9

[Guideline] HTTP Server Endpoints shall be capable of providing at least one media transport HTTP connection and it shall also be capable of processing sequential HTTP requests without responding with an HTTP error response code 503 (Service Unavailable).

[Attributes]

M	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	W3DTU

NOTE: This guideline ensures that rendering endpoints are capable of rendering a content binary using one media transport HTTP connection (See 7.5.4.3.2.4.4).

This guideline is not subject to preventing Content Sources to respond with 503 (Service Unavailable) when it currently is unable to process a HTTP request (See 7.5.4.3.2.4.2). But a Content Source will never return 503 (Service Not Available) under "Test Conditions".

7.5.4.3.2.4.10

[Guideline] An HTTP Server Endpoint shall use the HTTP status code of 503 (Service Unavailable) for an HTTP request only on the conditions that the Content Source does not have sufficient platform resources (network sockets, stored file in readable state, available tuner hardware, etc.) for sending the response with content byte data. On other conditions, the HTTP Server Endpoint shall not use the HTTP status code of 503 (Service Unavailable) for an HTTP request.

[Attributes]

M	R	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	FHYQ8

NOTE: This guideline defines the maximum response time for an HTTP Server Endpoint for a request for content.

In conjunction with 7.5.4.3.2.4.9, these guidelines ensure HTTP transactions for media transport.

Also, time-out value is for the worst case. HTTP Server Endpoints are urged to respond to a request as soon as possible.

7.5.4.3.2.4.11

[Guideline] An HTTP Client may treat HTTP Status code 503 (Service Unavailable) as equivalent to HTTP Status code 500 (Internal Server Error).

[Attributes]

O	C	DMP DMR DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	[33]	M8EP7

NOTE: The guideline clarifies a Content Receiver behavior for response message 503 with or without Retry-After. Hence, no retry is required.

7.5.4.3.2.5 MT HTTP Header Tolerance

7.5.4.3.2.5.1

[GUIDELINE] HTTP Client and Server Endpoints shall be tolerant of unknown HTTP headers.

Tolerant behavior is defined as being able to successfully "parse and interpret" or "parse and ignore" the HTTP headers and their values.

[ATTRIBUTES]

M	R	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[21] [33]	8EP7N

NOTE: This guideline addresses forward compatibility and allows for broader interoperability with implementations that employ transport layer vendor extensions by way of HTTP headers.

7.5.4.3.2.5.2

[GUIDELINE] Each HTTP header line (including the header's name and value but excluding the last carriage-return/line-feed sequence, CRLF) shall not exceed 998 bytes.

[ATTRIBUTES]

M	R	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33] [34]	SFHYQ

NOTE: These guidelines limit the length of an HTTP header line according to subclause 2.2.1 of [34]. The guidelines also specify the normative way to encode HTTP headers that span multiple lines.

Multi-line HTTP headers are always split at LWS characters.

7.5.4.3.2.5.3

[GUIDELINE] If an HTTP header line (header's name and value but excluding the last CRLF) exceeds 998 bytes, then the HTTP header shall span multiple lines. HTTP headers that span multiple lines shall prefix the additional lines with at least one space (SP) or horizontal tab (HT) as described in subclause 4.2 of [33].

[ATTRIBUTES]

M	R	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33] [34]	9GK5S

7.5.4.3.2.6 MT HTTP Header Case-Sensitivity

[GUIDELINE] Names of HTTP headers shall be treated as case insensitive tokens.

[ATTRIBUTES]

M	R	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[21] [33]	FW6L8

NOTE: This is normative according to the HTTP specification.

7.5.4.3.2.7 MT Baseline Transport: HTTP Server Endpoints

7.5.4.3.2.7.1

[GUIDELINE] HTTP Server Endpoints used for media transport purposes shall be compliant with HTTP/1.1, which also requires HTTP/1.0 compliance.

[ATTRIBUTES]

M	L	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	2EZ3F

NOTE: HTTP servers need to support both HTTP/1.0 and HTTP/1.1 requests to ensure wide interoperability.

7.5.4.3.2.7.2

[GUIDELINE] HTTP/1.1 Server Endpoints used for media transport should return HTTP version 1.1 in the response header, regardless of the version specified in the HTTP client's request.

[ATTRIBUTES]

S	R	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[23]	ZFW6L

NOTE: The clarifying RFC ([23]) clarifies that HTTP/1.1 servers are expected to return HTTP/1.1 even if the HTTP server receives a request marked with HTTP/1.0. The robustness rules, specified by the HTTP specification, enables clients and servers that employ different HTTP version numbers to coexist properly.

Also note that message format refers to both the HTTP headers and HTTP response body. As described by [23], the version field in a response message header indicates the protocol level that the server is capable of understanding. However, a server that understands protocol version 1.1 can generate messages compatible with version 1.0 in order to communicate with clients capable of handling only the lower 1.0 version. When this happens, the response message has a version header equal to 1.1 but the format of the message contains only version 1.0 headers. Reference [23] provides more details for interoperability between hosts with different HTTP versions.

7.5.4.3.2.7.3

[GUIDELINE] When responding to a request of version 1.0, an HTTP Server Endpoint shall format the response message in such a way that the result of decoding and processing the message does not depend on headers outside the scope of the HTTP 1.0 specification.

[ATTRIBUTES]

M	R	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[23] [33]	GK5S4

7.5.4.3.2.7.4

[GUIDELINE] HTTP Server Endpoints shall not report a higher version of HTTP than is actually supported by the implementation.

[ATTRIBUTES]

M	R	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	EZ3F6

7.5.4.3.2.7.5

[GUIDELINE] Interoperability between HTTP Client and Server Endpoints that implement different versions of the HTTP protocol shall follow the provisions and recommended actions defined in [23].

[ATTRIBUTES]

M	R	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[23]	Z3F6V

NOTE: Reference [23] defines the significance of HTTP version numbers, the rules for interoperability between hosts with different version numbers, and rules for the actual version number to be included when creating messages.

7.5.4.3.2.7.6

[GUIDELINE] HTTP Server Endpoints should support persistent HTTP/1.1 connections and pipelined HTTP/1.1 requests.

[ATTRIBUTES]

S	R	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	3F6V6

NOTE: The default behavior for HTTP servers responding to HTTP/1.1 requests is to support persistent connections, which means that the HTTP server can respond to multiple HTTP/1.1 requests on one HTTP session. Pipelined requests can be used to facilitate seek operations. See 7.5.4.3.6.5 for more information.

7.5.4.3.2.8 MT HTTP Header: scmsFlag.dlna.org

7.5.4.3.2.8.1

[GUIDELINE] HTTP Client and Server Endpoints may use the scmsFlag.dlna.org HTTP header, which indicates copyright assertion and copy status flags when transporting audio only content. HTTP Server Endpoints that serve DLNA media content and non-DLNA media content may also use this flag for the latter. HTTP Client Endpoints that encounter this HTTP header may implement behavior to enforce regional copyright provisions.

[ATTRIBUTES]

O	A	DMS DMP +PU+ +UP+ +DN+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33] [74]	L8OUB

NOTE: These guidelines make it possible to comply with regional legal requirements, such as in [74]. The flag is to be used with both DLNA and non DLNA audio only content. The syntax of the value is strictly defined by these guidelines.

7.5.4.3.2.8.2

[GUIDELINE] The notation of scmsFlag.dlna.org header field is defined as follows.

- scmsFlag.dlna.org = "scmsFlag.dlna.org" *LWS ":" *LWS flagValue
- flagValue = "00" | "01" | "10" | "11"

The value of the scmsFlag.dlna.org header shall be a two letter string from the following list: "00", "01", "10" or "11". The first and second characters can be set to 0 or 1 independently according to the rules below:

Definition of the value of the 1st character (i.e. left most) of the scmsFlag.dlna.org HTTP header:

- 0: copyright is asserted
- 1: no copyright is asserted

Definition of the value of the 2nd character of scmsFlag.dlna.org HTTP header:

- 0: Original recording
- 1: First generation or higher recording

The following example means copyright is asserted and first-generation or higher recording.

- scmsFlag.dlna.org : 01

[ATTRIBUTES]

M	A	DMS +PU+ +UP+ +UPSYNC+	M-DMS M-DMU	MIU	[33] [74]	F6V65

7.5.4.3.2.9 MT HTTP Header: Content-Type

7.5.4.3.2.9.1

[GUIDELINE] HTTP Server Endpoints shall specify the Content-Type HTTP header in the HTTP response header fields whenever it returns a Target Response to an HTTP GET operation.

[ATTRIBUTES]

M	L	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	S4NTS

NOTE: It is imperative that endpoints specify the Content-Type field to allow the receiver to know the MIME TYPE for the content that is to be sent in the HTTP response message.

Also note that MIME-TYPE values appear in the Content-Type HTTP header.

7.5.4.3.2.9.2

[GUIDELINE] The MIME-TYPE values that appear in clause 5 of [56] shall be used as values for Content-Type when an HTTP message describes DLNA media contents.

[ATTRIBUTES]

M	L	DMS +PU+ +PR1+ +PR2+ +UP+ +UPSYNC+	M-DMS M-DMU	MIU	[56]	K5S4N

NOTE: This guideline specifies the correct mime type values for use with the Content-Type header field when transporting content encoded in a DLNA media format.

7.5.4.3.2.9.3

[GUIDELINE] HTTP Client Endpoints shall specify the Content-Type HTTP header in the HTTP request if using a POST operation to send data.

[ATTRIBUTES]

M	L	+UP+ +UPSYNC+	M-DMU	MIU	[33]	6L8OU

7.5.4.3.2.10 MT HTTP Header: contentFeatures.dlna.org

7.5.4.3.2.10.1

[GUIDELINE] If an HTTP Server Endpoint receives an HTTP GET or HEAD request with the getcontentFeatures.dlna.org HTTP header, then the HTTP server shall use the contentFeatures.dlna.org HTTP header if it responds with a Target Response.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33] [49]	7NKR8

NOTE: As noted, this guideline permits an HTTP Server endpoint o use the contentFeatures.dlna.org in an HTTP response.

7.5.4.3.2.10.2

[GUIDELINE] An HTTP Server Endpoint may respond with the contentFeatures.dlna.org HTTP header to an HTTP GET or HEAD request that does not have the getcontentFeatures.dlna.org HTTP header.

[ATTRIBUTES]

O	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33] [49]	5S4NT

7.5.4.3.2.10.3

[GUIDELINE] HTTP Client Endpoints may use the getcontentFeatures.dlna.org when issuing GET or HEAD requests.

[ATTRIBUTES]

O	A	DMP DMR DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	[33] [49]	Q868P

NOTE: These guidelines describe how an HTTP client endpoint can request an HTTP server to use the contentFeatures.dlna.org in the response.

7.5.4.3.2.10.4

[GUIDELINE] The notation of getcontentFeatures.dlna.org header field is defined as follows:

- getcontentFeatures.dlna.org = "getcontentFeatures.dlna.org" *LWS ":" *LWS "1"

The only value possible is "1".

Example:

- getcontentFeatures.dlna.org: 1

[ATTRIBUTES]

M	A	DMS DMP DMR DMPr +DN+ +UP+ +PU+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33] [49]	W6L8O

7.5.4.3.2.10.5

[Guideline] If an HTTP Server Endpoint receives any value except "1" in the getcontentFeatures.dlna.org header it shall return an error code response of 400 (Bad Request).

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	[33] [49]	EP7NK

7.5.4.3.2.10.6

[Guideline] The value of the contentFeatures.dlna.org HTTP header shall be the same value as the fourth field of the content's res@protocolInfo value, as described in the 7.4.1.3.16 MM protocolinfo values: 4th Field guideline.

The notation of contentFeatures.dlna.org header field for DLNA media transport is defined as follows;

- contentFeatures-line = "contentFeatures.dlna.org" *LWS ":" *LWS 4th-field
- 4th-field = <case sensitive 4th field value defined in guideline 7.4.1.3.16.1>

[Attributes]

M	A	DMS +PU+ +UP+ +PR1+ +PR2+ +UPSYNC+	M-DMS M-DMU	MIU	[33] [49]	TU5R3

NOTE: This guideline allows HTTP transport transactions to carry information (that is normally only accessible at the UPnP AV ContentDirectory service layer) about the requested content (and the server capabilities for that content).

Note that this header can be used by Content Sources for non-DLNA content.

7.5.4.3.2.10.7

[Guideline] HTTP Client Endpoints shall not include the following headers in HTTP requests, unless the appropriate protocolInfo 4th field parameters indicate support for them.

- Range
- TimeSeekRange.dlna.org
- PlaySpeed.dlna.org
- getAvailableSeekRange.dlna.org

[Attributes]

M	A	DMP DMR DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	[33] [49]	P7NKR

NOTE: The 4th field of a protocolInfo is obtained from contentFeatures.dlna.org. Devices can also directly check metadata in the CDS, without using contentFeatures.dlna.org.

These guidelines do not define interoperability guidelines for a scenario where an HTTP Client Endpoint attempts to use an optional transport layer feature when the 4th field does not indicate support for the transport layer feature.

7.5.4.3.2.10.8

[Guideline] An HTTP client that issues a POST request for uploading DLNA-conformant content shall use the contentFeatures.dlna.org HTTP header. The value sent shall match the

4th field that was sent in the CDS:CreateObject request, as defined in 7.4.1.7.19.6 (MM/CM: General Rule for Creating <res> Elements that Support a Content Transfer Process).

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[33] [49]	YQ868

NOTE: These guidelines explain how contentFeatures.dlna.org is used in POST transactions.

7.5.4.3.2.10.9

[GUIDELINE] If an HTTP Server Endpoint receives a POST request

- without the contentFeatures.dlna.org HTTP header or
- with a DLNA.ORG_PN parameter that is inconsistent with guideline 7.5.4.3.2.10.8

then the HTTP Server Endpoint may respond with an HTTP error.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[33] [49]	U5R3P

NOTE: This guideline allows an HTTP Server Endpoint to accept upload of non-DLNA content or allow it to reject such content.

7.5.4.3.2.11 MT HTTP Header: dlna pragma-directive (ifoFileURI.dlna.org)

7.5.4.3.2.11.1

[GUIDELINE] If an HTTP Server Endpoint provides a res@dlna:ifoFileURI for a resource (see 7.4.1.4.8) and the HTTP Server Endpoint receives an HTTP GET or HEAD request with the Pragma-with-getIfoFileURI-pragma-directive in the HTTP request, then the HTTP Server Endpoint shall provide the Pragma-with-ifoFileURI-pragma-directive (with the res@dlna:ifoFileURI value associated with the target URI) if it responds with a Target Response.

If an HTTP Server Endpoint does not provide a res@dlna:ifoFileURI for a resource, then the HTTP server shall not provide the Pragma-with-ifoFileURI-pragma-directive in the HTTP response.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33] [49]	HYQ86

NOTE: subclause 14.32 of [33] defines the Pragma as a general-header for implementation-specific directives.

The DLNA.ORG_PN parameter in the contentFeatures.dlna.org header field indicates the DLNA media format profile ID. Furthermore, as described in 7.4.1.4.8, content profiled as MPEG_PS_NTSC or MPEG_PS_PAL can have an IFO file. If the HTTP response to a request with the Pragma-with-getIfoFileURI-pragma-directive does not include the Pragma-with-ifoFileURI-pragma-directive, then the HTTP client endpoint needs to be aware that the AV content does not have an IFO file.

Implementers need to be careful not to exceed the maximum 998 byte limit for HTTP header lines when using the Pragma-with-IfoFileURI-pragma-directive.

7.5.4.3.2.11.2

[GUIDELINE] HTTP Client Endpoints may use the Pragma-with-getIfoFileURI-pragma-directive when issuing GET or HEAD requests.

[ATTRIBUTES]

O	A	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33] [49]	R368A

7.5.4.3.2.11.3

[GUIDELINE] The notation of the PRAGMA header field with the getIfoFileURI-pragma-directive is defined as follows:

- Pragma-with-getIfoFileURI-pragma-directive = "PRAGMA" *LWS ":" *LWS *(pragma-directive *LWS "," *LWS) getIfoFileURI-pragma-directive *(*LWS "," *LWS pragma-directive)
- pragma-directive = "no-cache" | extension-pragma
- extension-pragma = token ["=" (token | quoted-string)]
- getIfoFileURI-pragma-directive = "getIfoFileURI.dlna.org"

Note that the PRAGMA header name, pragma-directive token, and the getIfoFileURI-pragma-directive token are case insensitive.

Examples:

- PRAGMA: getIfoFileURI.dlna.org
- PRAGMA: getIfoFileURI.dlna.org, no-cache

[ATTRIBUTES]

M	A	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33] [49] [98]	QWR36

NOTE: The "1#token" syntax means a comma separated list including one or more elements. LWS can appear before and after the separator comma ',' in this syntax. See subclause 2.1 of [33].

Note that the Pragma-with-getIfoFileURI-pragma-directive can be used by endpoints issuing an HTTP request on non-DLNA content.

7.5.4.3.2.11.4

[GUIDELINE] The notation of the PRAGMA header field with the ifoFileURI-pragma-directive is defined as follows

- Pragma-with-ifoFileURI-pragma-directive = "PRAGMA" *LWS ":" *LWS *(pragma-directive *LWS "," *LWS) ifoFileURI-pragma-directive *(*LWS "," *LWS pragma-directive)
- pragma-directive = "no-cache" | extension-pragma
- extension-pragma = token ["=" (token | quoted-string)]
- ifoFileURI-pragma-directive = "ifoFileURI.dlna.org" "=" quoted-absolute-uri-string
- quoted-absolute-uri-string = <same value as the corresponding res@dlna:ifoFileURI attribute value, as described in 7.4.1.4.8. It shall be quoted by """.>

Example:

- PRAGMA: ifoFileURI.dlna.org="http://192.168.0.1:8080/IFO_101.ifo"

The PRAGMA HTTP header name, the pragma-directive token, and the ifoFileURI-pragma-directive token are case insensitive.

A Content Source that provides the Pragma-with-ifoFileURI-pragma-directive with non-DLNA content shall also provide an IFO file that conforms to the guidelines 9.3.2.7 through 9.3.2.9 in [56].

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33] [49] [56]	DTU5R

7.5.4.3.2.12 MT HTTP HEAD Requests

7.5.4.3.2.12.1

[GUIDELINE] HTTP Server Endpoints (HTTP/1.1 and HTTP/1.0) shall respond to HTTP HEAD requests, using the guidelines described below.

[ATTRIBUTES]

M	R	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	WR368

NOTE: There are several interpretations to the format of HEAD responses. These guidelines provide consistent interpretation.

7.5.4.3.2.12.2

[GUIDELINE] A Target Response to a HEAD request shall be composed of only HTTP headers and a zero-length response entity body.

[ATTRIBUTES]

M	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	RWS43

7.5.4.3.2.12.3

[GUIDELINE] Target Responses to identical HTTP version HEAD and GET requests for a given content binary shall use consistent transfer encoding headers. For example, if Content-Length is included in a HEAD Target Response then Content-Length shall also be present in the GET Target Response for the same content binary.

This guideline assumes that the content binary has not changed between the different HTTP requests.

[ATTRIBUTES]

M	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	S4397

NOTE: This guideline forbids HTTP Server Endpoints from responding to a GET request using a different encoding method than the HEAD Target Response. E.g. A DMS could not respond to a HEAD request with Content-Length and respond to a GET request using chunked encoding.

7.5.4.3.2.12.4

[GUIDELINE] If an HTTP server does not know the length of a requested resource, such as in the case when *Chunked Transfer Coding* is employed in HTTP/1.1, the Content-Length field shall be omitted from the HEAD Target Response.

[ATTRIBUTES]

M	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	WS439

7.5.4.3.2.12.5

[GUIDELINE] If an HTTP Server Endpoint (HTTP/1.1 and HTTP/1.0) responds to an HTTP GET request with a non-error response, the HTTP server should respond to the equivalent HEAD request with a non-error response.

A successful response is defined as an HTTP response with a status code in the 1xx or 2xx range.

[ATTRIBUTES]

S	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	ZS4J4

NOTE: HTTP specification requires HTTP servers to respond to HEAD requests. This guidelines clarifies that HTTP servers cannot respond with an error code for HEAD requests that target a valid URI for the HTTP server.

This is not mandatory because conditions can be different than for those of the GET request (e.g. server saturation, etc).

7.5.4.3.2.12.6

[GUIDELINE] The HTTP headers of a HEAD Target Response shall include the Content-Type HTTP header.

[ATTRIBUTES]

M	L	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	S4J4Y

NOTE: Ideally, an HTTP server can know all of the HTTP headers (for requests that use TimeSeekRange.dlna.org, Range, PlaySpeed.dlna.org, or other HTTP headers) that will be sent in an HTTP response without doing any of the computational work to buffer content data, but some scenarios (such as those involving transcoding, live streams, random access requests, etc.) require a lot of computational cycles. For these reasons, these guidelines specify minimal expectations for HEAD responses while recommending the ideal expectations.

7.5.4.3.2.12.7

[GUIDELINE] HTTP HEAD Target Responses should have the exact same HTTP headers as those in the equivalent HTTP GET Target Response.

[ATTRIBUTES]

S	R	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	UZS4J

7.5.4.3.2.12.8

[GUIDELINE] An HTTP client should not issue an HTTP HEAD request to a URI intended for a file upload (e.g. res@importUri, res@dlna:importIfoFileURI).

[ATTRIBUTES]

S	C	+UP+ +UPSYNC+	M-DMU	MIU	[33]	V6QW5

NOTE: The MediaServers behavior of a HEAD request is undefined because implementations that use the same URI value for res@importUri and the <res> value might choose to only respond in the manner of a GET request.

7.5.4.3.2.13 MT Image File Size Acquisition via HTTP HEAD

7.5.4.3.2.13.1

[GUIDELINE] An HTTP Server that serves an image content binary should respond to HTTP/1.0 HEAD requests with the Content-Length header to indicate the length of the image content binary.

[ATTRIBUTES]

S	L	DMS +PU+ +PR1+	M-DMS	n/a	[33]	XV6QW	

NOTE: HTTP/1.0 HEAD requests allow control points to query a server for the length of an image file, when res@size metadata is not available.

The reason why HTTP/1.0 is used for this purpose is that chunked transfer coding and Content-Length are used in a mutually exclusive manner. Since HTTP servers are required to support HTTP/1.1, many servers use chunked transfer coding. Furthermore, DLNA guidelines require HTTP headers to be the same for equivalent GET and HEAD requests. Therefore, using HTTP/1.0 for the HEAD request improves the odds of success.

Note that control points that fail to acquire the file size in this manner are still governed by 7.4.2.3.5.4 (MM XHTML-Print), meaning that a control point that accidentally specifies a page that exceeds the permitted image byte total is in violation of the guidelines.

7.5.4.3.2.13.2

[GUIDELINE] A Printing control point that attempts to acquire the length of an image content binary by using an HTTP HEAD request should use an HTTP/1.0 HEAD request.

[ATTRIBUTES]

S	L	+PR2+	n/a	n/a	[33]	QDWQ3	

7.5.4.3.2.14 MT HTTP Header Parsing (Server)

[GUIDELINE] HTTP Server Endpoints shall gracefully skip over unsupported HTTP header fields. Under no circumstances can an HTTP server fail to process a properly formatted HTTP request because of an unrecognized or unsupported HTTP header field.

[ATTRIBUTES]

M	R	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	LZU6X	

NOTE: Incorrect HTTP header parsing has been the source of numerous compatibility issues during plugfest events.

7.5.4.3.2.15 MT HTTP Header: Content-Length

7.5.4.3.2.15.1

[GUIDELINE] If the Content-Length HTTP header is present in an HTTP message with a message body, then the message shall not use *Chunked Transfer Coding*.

[ATTRIBUTES]

M	C	DMS +PU+ +UP+ +UPSYNC+	M-DMS M-DMU	MIU	[33]	WXV6Q	

NOTE: The usage of Content-Length is not allowed under subclause 4.4 of the HTTP spec in [33]

Applies to both HTTP GET response messages and POST request messages.

7.5.4.3.2.15.2

[GUIDELINE] If the Content-Length HTTP header is omitted from an HTTP GET response, then the HTTP Server Endpoint shall do one of the following.

- The HTTP server will close the TCP connection after sending the last byte of the response message. Furthermore, if the HTTP server is responding to an HTTP/1.1 transaction, then the HTTP server shall also use the CONNECTION: CLOSE header and value to explicitly indicate that it will close the connection. Lastly, any additional byte sequence following the headers shall be treated as entity-body bytes until the instant when the connection is closed.

- The HTTP/1.1 server will use chunked transfer-coding for the response when communicating with an HTTP/1.1 client.

This guideline applies in all scenarios with the following exceptions:

- Response messages that are prohibited from having an entity-body (such as the 1xx, 204, and 304 responses).
- The HTTP server returns an HTTP/1.1 response with no entity-body, *Chunked Transfer Coding* is not used, and the CONNECTION:CLOSE header is not used (i.e. persistent connection).

[ATTRIBUTES]

M	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	VTLZU	C

NOTE: These guidelines clarify the expected behavior regarding Content-Length usage for response messages.

For pipelined requests, if the server decides to close a TCP connection for some response, any additional requests submitted afterwards will not be processed by the server.

If the Content-Length is used in messages with no entity body, then the accurate value of "0" is required per guideline 7.5.4.3.2.15.5. Likewise messages encoded with *Chunked Transfer Coding* will use the accurate value of "0" for the chunk-size.

7.5.4.3.2.15.3

[GUIDELINE] If the Content-Length HTTP header is omitted from an HTTP POST request, then the HTTP Client Endpoint shall use *Chunked Transfer Coding* for the request.

[ATTRIBUTES]

M	C	+UP+ +UPSYNC+	M-DMU	MIU	[33]	ZLQDW	

NOTE: See subclause 4.4 of [33]

7.5.4.3.2.15.4

[GUIDELINE] An HTTP message that carries no content binary data in the entity body, and which uses *Chunked Transfer Coding* shall use a single chunk with chunk-size indicator equal to 0.

[ATTRIBUTES]

M	R	DMS +PU+ +UP+ +UPSYNC+	M-DMS M-DMU	MIU	[33]	LQDWQ	

NOTE: Some response messages do not carry content binary data in the message payload. If the HTTP Server Endpoint is using *Chunked Transfer Coding* to produce this message, then the message has a single one-line chunk-size indicator with a value of 0.

7.5.4.3.2.15.5

[GUIDELINE] When operating under persistent connections (including pipelining), an HTTP/1.1 client shall detect the existence of a message entity body when it receives a message with:

- A non-zero Content-Length header.
- Non-zero chunk-size values when using *Chunked Transfer Coding*.
- Non-zero content bytes following the message headers when the HTTP server declares that it will close the connection. (An HTTP server declares that it will close the connection by using the "CONNECTION: CLOSE" header in the response.)

If the Content-Length header and the CONNECTION:CLOSE header are not provided and *Chunked Transfer Coding* is not used in the HTTP/1.1 response, then the message has no entity body.

[ATTRIBUTES]

M	C	DMP DMR DMPr +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33]	TLZU6	

NOTE: This guideline clarifies the process used by a client to parse and extract the body (if any) of received response messages.

Notice that response messages that do not carry a Content-Length, and do not use transfer encoding, could carry content bytes if the server closes the connection after sending the last byte. The server needs to explicitly announce that the connection will be closed by using the adequate header.

7.5.4.3.2.15.6

[GUIDELINE] When operating under persistent connections (including pipelining), an HTTP/1.1 Server shall detect the existence of a message entity body when it receives a POST message with:

- A non-zero Content-Length header.
- Non-zero chunk-size values when using *Chunked Transfer Coding*.

[ATTRIBUTES]

M	C	DMS	M-DMS	n/a	[33]	6XJSB	

NOTE: This guideline clarifies the process used by a server to parse and extract the body (if any) of a request messages.

7.5.4.3.2.15.7

[GUIDELINE] If an HTTP Server Endpoint knows the byte length of a response body, then the HTTP server should use the Content-Length HTTP header in the HTTP Target Response.

[ATTRIBUTES]

S	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	ZU6XJ	

NOTE: As a general rule, the Content-Length provides useful information to HTTP clients. However, the Content-Length HTTP header is not required because it is difficult to provide an accurate byte length in some scenarios. For example, HTTP servers that are transmitting live content (and some transcoded content) might not know the value for the Content-Length HTTP header field. In cases when the Content-Length is provided, the value needs to match the byte length of the response body.

7.5.4.3.2.15.8

[GUIDELINE] If the HTTP Server Endpoint does not know the byte length of the response entity body or if Content-Length cannot be used due to some exceptions listed in [33] subclause 4.4, then HTTP servers shall omit the Content-Length HTTP header from HEAD, and GET Target Responses.

[ATTRIBUTES]

M	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	U6XJS	

7.5.4.3.2.15.9

[GUIDELINE] If an HTTP Server Endpoint sends an HTTP GET Target Response with the Content-Length HTTP header, then the byte length of the response entity body shall match the value of the Content-Length HTTP header.

[ATTRIBUTES]

M	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	DWQ3K

7.5.4.3.2.15.10

[GUIDELINE] If an HTTP Server Endpoint receives an HTTP/1.0 GET request and sends an HTTP Target Response that does not have a Content-Length HTTP header, then the HTTP server shall close the TCP connection when all data is transferred.

[ATTRIBUTES]

M	R	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	QW58V

NOTE: The HTTP server is required to close the TCP connection in this scenario because only the HTTP server knows when it has finished sending the bytes for the requested URI.

7.5.4.3.2.15.11

[GUIDELINE] If an HTTP Client Endpoint knows the byte length of the entity-body in a POST request, then the HTTP client should use the Content-Length HTTP header in the HTTP request.

[ATTRIBUTES]

S	R	+UP+ +UPSYNC+	M-DMU	MIU	[33]	Q3KYV

7.5.4.3.2.15.12

[GUIDELINE] If the HTTP Client Endpoint does not know the byte length of the entity-body in a POST request or if Content-Length cannot be used, the HTTP Client Endpoint shall omit the Content-Length HTTP header from the POST request and use the *Chunked Transfer Coding* as specified in [33] subclause 4.4.

[ATTRIBUTES]

M	R	+UP+ +UPSYNC+	M-DMU	MIU	[33]	6QW58

7.5.4.3.2.15.13

[GUIDELINE] If an HTTP Client Endpoint sends an HTTP POST Request with the Content-Length HTTP header, then the byte length of the entity-body shall match the value of the Content-Length HTTP header.

[ATTRIBUTES]

M	R	+UP+ +UPSYNC+	M-DMU	MIU	[33]	WQ3KY

7.5.4.3.2.16 MT Maximum Byte Size Transfers

7.5.4.3.2.16.1

[GUIDELINE] HTTP Client and Server Endpoints shall not use values that exceed 281474976710655 (i.e. 2^48 - 1) for the following HTTP fields.

- Content-Length header
- first byte pos and last byte pos (as defined in subclauses 14.35.1 and 14.16 of [33] and guideline 7.5.4.3.2.22.3)
- instance-length (as defined in subclause 14.16 of [33])
- chunk-size (as defined in subclause 3.6.1 of [33])

[Attributes]

M	L	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33]	4J4YC

NOTE: The HTTP specification ([33]) does not limit the maximum content length. Note that a 32 bit integer is not sufficient, especially, for a 2 hour MPEG-2 stream that exceeds 4 GBytes. The specified range covers the maximum size of a DLNA content binary.

Please note that chunk-size is in hexadecimal form, while the other fields are in decimal form.

7.5.4.3.2.16.2

[Guideline] An HTTP Client or Server Endpoint shall parse and interpret values up to 281474976710655 (i.e. 2^48 - 1) for the Content-Length header field, Range Units in bytes, and chunk-size field that are represented in HTTP requests/responses.

[Attributes]

M	L	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33]	W58V5

7.5.4.3.2.17 MT HTTP/1.0 Persistent Connections (Server)

[Guideline] This Guideline no longer applies.

[Attributes]

		n/a	n/a	n/a	n/a	4YC9L

7.5.4.3.2.18 MT HTTP Common Random Access Data Availability Requirements

7.5.4.3.2.18.1

[Guideline] If an HTTP Server Endpoint supports the Range header field or TimeSeekRange.dlna.org header fields for a content binary, it should support and process an HTTP HEAD request to get the current random access data range for the content binary while it is processing the HTTP GET request to transmit the Target Response.

The process for acquiring the random access data ranges for "Full Random Access Data Availability" and "Limited Random Access Data Availability" model are described in 7.5.4.3.2.19 and 7.5.4.3.2.20, respectively.

All guidelines in 7.5.4.3.2.18 apply for both "Full Random Access Data Availability" and "Limited Random Access Data Availability"models.

[ATTRIBUTES]

S	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	J4YC9

NOTE: Guideline 7.5.4.3.2.4.4 states that HTTP clients will not assume that HTTP servers accept more than one media transport HTTP connection at a time. Although an HTTP client will not assume that the HTTP server accepts more than one HTTP GET request simultaneously, this guideline clarifies the intent for HEAD requests. In cases where a random access data range is updated while an HTTP client is receiving streaming data from a HTTP GET request, it is useful for the HTTP client to use an HTTP HEAD request to get the current random access data range simultaneously.

7.5.4.3.2.18.2

[GUIDELINE] If an HTTP Server Endpoint supports the TimeSeekRange.dlna.org header field for the specified URI, it shall process any requested range in the random access data range at the time.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	97ES2

NOTE: This is a general requirement that random access requests be supported on the entire [r_0, r_N] data range.

7.5.4.3.2.18.3

[GUIDELINE] If an HTTP Server Endpoint supports the Range header field for the specified URI, it shall process any requested range in the random access data range at the time.

[ATTRIBUTES]

M	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	397ES

NOTE: This is a general requirement that random access requests be supported on the entire [r_0, r_N] data range.

7.5.4.3.2.19 MT HTTP Data Range of "Full Random Access Data Availability"

7.5.4.3.2.19.1

[GUIDELINE] When using the "Full Random Access Data Availability" model, the npt value of 0 and the byte position of 0 for the TimeSeekRange.dlna.org header field shall refer to the beginning of the content binary (i.e. the first byte available for the content binary).

All guidelines in 7.5.4.3.2.19 apply only in the case of "Full Random Access Data Availability" models.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+ +DN+ +DNSYNC+	M-DMS M-DMP M-DMD	MIU	[33]	4397E

NOTE: These guidelines clarify guidelines 7.5.4.2.15 MT "Full Random Access Data Availability" Model by explaining how npt values and byte indices are used with the TimeSeekRange.dlna.org and Range headers.

7.5.4.3.2.19.2

[GUIDELINE] If the end position is not specified in a GET request with Range or TimeSeekRange.dlna.org, then the HTTP Server Endpoint's transmission of the Target Response shall include the content data that is currently available and the content data that will be available in the future for the current stream. (i.e. Assuming s_N is changing with time, the server keeps transmitting data until s_N becomes permanently fixed.)

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	8A57U

7.5.4.3.2.19.3

[GUIDELINE] The byte position of 0 for the Range header field shall refer to the beginning of the content binary.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+ +DN+ +DNSYNC+	M-DMS M-DMP M-DMD	MIU	[33]	3PU8U

7.5.4.3.2.19.4

[GUIDELINE] If an HTTP Server Endpoint receives an HTTP GET request that omits the Range and TimeSeekRange.dlna.org HTTP headers, then the HTTP Server Endpoint shall infer a byte-pos of 0 (i.e. respond from the beginning of the content binary).

[ATTRIBUTES]

M	A	DMS	M-DMS	MIU	[33]	368A5

NOTE: This guideline ensures that HTTP clients will receive content in a manner consistent with the conventions established by HTTP.

7.5.4.3.2.19.5

[GUIDELINE] If an HTTP Server Endpoint supports the Range HTTP header (as defined in the 7.5.4.3.2.22 MT HTTP Header: Range (Server)) with the "Full Random Access Data Availability" model, the HTTP Server Endpoint should specify the instance-length in the Content-Range HTTP header field of the Target Response to a HTTP GET/HEAD request with the Range HTTP header field.

[ATTRIBUTES]

S	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	R3PU8

NOTE: This guideline clarifies the subclause 14.16 of [33]. The asterisk "*" can be used instead of the instance-length if the instance-length is unknown at the time when the response was generated or when s_N-increasing is true. However DLNA recommends that HTTP servers provide an instance-length.

7.5.4.3.2.19.6

[GUIDELINE] If an HTTP Client Endpoint wants to get the instance-length of a content binary, then it should use an HTTP HEAD request with the Range HTTP header that specifies 0 for the first-byte-position and omits the end byte-position.

[ATTRIBUTES]

S	C	DMP DMR DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	[33]	68A57

NOTE: The response will include the Content-Range header. If the instance length is known, then it will be number of bytes in the UCDAM [s_0, s_N] data range. Otherwise, the instance-length token will be "*".

7.5.4.3.2.19.7

[GUIDELINE] If an HTTP Server Endpoint is serving stored content it shall provide the instance-length in the Content-Range header.

[ATTRIBUTES]

M	A	DMS +PU+ +UPSYNC+	M-DMS	n/a	[33]	5R3PU	

7.5.4.3.2.19.8

[GUIDELINE] If an HTTP Server Endpoint supports the TimeSeekRange.dlna.org HTTP header (as defined in guideline 7.5.4.3.2.24) with the "Full Random Access Data Availability" model, the HTTP Server Endpoint should specify the instance-duration in the TimeSeekRange.dlna.org HTTP header field of the Target Response to an HTTP GET/HEAD request with the TimeSeekRange.dlna.org HTTP header field.

[ATTRIBUTES]

S	C	DMS +PU+	M-DMS	n/a	[33]	8PW38	

NOTE: This guideline clarifies guideline 7.5.4.3.2.24.3. The syntax does not require an instance-duration, but DLNA recommends specifying the instance-duration.

7.5.4.3.2.19.9

[GUIDELINE] In conjunction with the guideline 7.5.4.3.2.19.8, if an HTTP Server Endpoint supports the TimeSeekRange.dlna.org HTTP header for a content binary, it should specify the instance-length in the TimeSeekRange.dlna.org HTTP header field of the Target Response to an HTTP GET/HEAD request with the TimeSeekRange.dlna.org HTTP header field.

[ATTRIBUTES]

S	C	DMS +PU+	M-DMS	n/a	[33]	6V65V	

NOTE: Even if TimeSeekRange.dlna.org is supported without the Range header, then the HTTP server is urged to provide an instance-length when possible.

Note that guideline 7.5.4.3.2.24.5 requires instance-length if the Range HTTP header field is also supported.

7.5.4.3.2.20 MT HTTP: Data Range of "Limited Random Access Data Availability"

7.5.4.3.2.20.1

[GUIDELINE] If the end position is not specified in a GET request with Range or TimeSeekRange.dlna.org, then the HTTP Server Endpoint's transmission of the Target Response shall include the content data that is currently available and the content data that will be available in the future for the current stream. (i.e. Assuming s_N is changing with time, the server keeps transmitting data until s_N becomes permanently fixed.)

All guidelines in 7.5.4.3.2.20 apply only in the case of "Limited Random Access Data Availability"models.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[33]	OUBZ3	

NOTE: In the case of live content, the end of the content binary is undefined. If the end position is not specified in the request, it means the client wants to continue receiving data until the absolute end of the stream.

7.5.4.3.2.20.2

[Guideline] Since the random access data range of [r_0, r_N] is able to change at any time, the HTTP Client Endpoint should not assume that a GET request (with either Range or TimeSeekRange.dlna.org) that specifies a range with a previous random access data range will always succeed.

[Attributes]

S	C	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33]	KR8WA

NOTE: An HTTP request with a range request can fail with an HTTP response error code of: 416 (Requested Range Not Satisfiable) due to changing random access data range.

7.5.4.3.2.20.3

[Guideline] If the HTTP Client Endpoint includes both the TimeSeekRange.dlna.org HTTP header field and the PlaySpeed.dlna.org HTTP header field in an HTTP GET request for a content binary in case of the "Limited Random Access Data Availability" model, it shall specify the end position of the request range in the range specifier of the TimeSeekRange.dlna.org HTTP header. This guideline applies to both positive speed value (forward scan mode) and negative speed value (backward scan mode).

[Attributes]

M	C	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33]	68PW3

NOTE: Even if the HTTP server supports the PlaySeed.dlna.org HTTP header field for server side's trick mode, it cannot process it beyond the current available data in case of "Limited Random Access Data Availability".

7.5.4.3.2.20.4

[Guideline] If an HTTP Server Endpoint supports either the Range HTTP header (as defined in the guideline 7.5.4.3.2.22) or the TimeSeekRange.dlna.org HTTP header (as defined in the guideline 7.5.4.3.2.24) with the "Limited Random Access Data Availability" model for a content binary, the HTTP Server Endpoint shall support the getAvailableSeekRange.dlna.org HTTP header field and availableSeekRange.dlna.org HTTP header field in HTTP GET/HEAD requests for the content binary.

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	[33]	868PW

NOTE: Essentially, a server that supports the "Limited Random Access Data Availability" model for HTTP needs to implement support for the getAvailableSeekRange.dlna.org and availableSeekRange.dlna.org HTTP headers. These headers allow clients to request the random access data range and allow servers to respond with that information.

7.5.4.3.2.20.5

[Guideline] If an HTTP Server Endpoint that supports the availableSeekRange.dlna.org HTTP header field (for a content binary) receives an HTTP HEAD/GET request with the getAvailableSeekRange.dlna.org HTTP header, it shall respond with the availableSeekRange.dlna.org HTTP header field to return the current random access data range.

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	[33]	NKR8W

7.5.4.3.2.20.6

[GUIDELINE] The notation of getAvailableSeekRange.dlna.org header field shall be as follows:

- getAvailableSeekRange-line = "getAvailableSeekRange.dlna.org" *LWS ":" *LWS "1"

The only value possible is "1". Any other values sent shall result in the HTTP server responding with an error code of 400 (Bad Request).

Example:

- getAvailableSeekRange.dlna.org: 1

[ATTRIBUTES]

M	A	DMS DMP DMR +DN+ +PU+ +DNSYNC+	M-DMS M-DMP M-DMD	MIU	[33]	R8WAA

7.5.4.3.2.20.7

[GUIDELINE] If an HTTP Server Endpoint receives any value except "1" in the getAvailableSeekRange.dlna.org header it shall return an error code response of 400 (Bad Request).

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	8OUBZ

7.5.4.3.2.20.8

[GUIDELINE] The notation of the availableSeekRange.dlna.org header field for DLNA media transport shall be as follows:

- availableSeekRange-line = "availableSeekRange.dlna.org" *LWS ":" *LWS mode-flag SP range specifier
- mode-flag = "0" | "1"
- range specifier = npt range [SP byte-range] | byte-range
- npt range = "npt" "=" npt time "-" npt time
- npt time = <syntax defined in 7.5.4.2.13 MT Normative Syntax for npt-time>
- byte range = "bytes" "=" first byte pos "-" last byte pos
- first byte pos = 1*DIGIT
- last byte pos = 1*DIGIT

Note that literals, "npt" and "bytes", are case sensitive.

Examples:

- availableSeekRange.dlna.org: 0 bytes=214748364-224077003
- availableSeekRange.dlna.org: 0 npt=00:05.30.12-00:10:34
- availableSeekRange.dlna.org: 1 npt=00:05.30.12-00:10:34 bytes=214748364-224077003

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	UBZ33

7.5.4.3.2.20.9

[GUIDELINE] If mode-flag is "0", then the following behaviors shall be implemented by the HTTP Server Endpoint when responding with the availableSeekRange.dlna.org HTTP header.

- The HTTP Server Endpoint shall implement rules in 7.5.4.2.16 MT "Limited Random Access Data Availability" Model, guideline 7.5.4.2.16.2 and guideline 7.5.4.2.16.3 for Mode=0.
- The range-specifier (i.e. npt-range and/or bytes-range) shall map to the [r_0, r_N] data range, such that if the HTTP Server Endpoint receives a request that specifies a bytes-range or npt-range that is inclusively within the server's range-specifier, then the HTTP Server Endpoint shall be able to serve the requested data bytes.
- If the HTTP Server Endpoint receives a GET request that omits both Range and TimeSeekRange.dlna.org it shall respond from the s_N data boundary, as defined in 7.5.4.2.16.2.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	NTSXZ

NOTE: This guideline clarifies the Mode=0 rules described in 7.5.4.2.16 MT "Limited Random Access Data Availability" Model, Requirement 7.5.4.2.16.2, by explaining how specific syntax tokens refer to the UCDAM data boundaries.

Essentially the npt and byte positions in the range-specifier indicate the valid data range for use in requests that use Range or TimeSeekRange.dlna.org. Similarly a request omits these headers results with content data from the live position.

7.5.4.3.2.20.10

[GUIDELINE] An HTTP Client Endpoint operating under the "Limited Random Access Data Availability" model shall specify a valid value for the first-byte-pos and/or the starting npt-time tokens in a request that specifies the Range and/or TimeSeekRange.dlna.org HTTP header.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[33]	V65V8

NOTE: As a corollary, a request that explicitly indicates "0" for npt or a byte position only works if the HTTP Server Endpoint indicates "0" is valid through the availableSeekRange.dlna.org.

7.5.4.3.2.20.11

[GUIDELINE] If mode-flag is '1', then the following behaviors shall be implemented by the HTTP Server Endpoint when responding with the availableSeekRange.dlna.org HTTP header.

- The HTTP Server Endpoint shall implement rules in 7.5.4.2.16.4 (MT "Limited Random Access Data Availability" Model) for Mode=1.
- If present, the npt-range shall map to [r_0, r_N] such that if the HTTP Server Endpoint receives a request that specifies an npt-range that is inclusively within the server's range-specifier (as indicated in availableSeekRange.dlna.org), then the HTTP Server Endpoint shall be able to serve the requested data bytes in a timely manner.
- If present, the bytes-range map to a subset of [r_0, r_N] such that if an HTTP Server Endpoint receives a request that specifies a bytes-range that is inclusively within the server's range-specifier (as indicated in availableSeekRange.dlna.org), then the HTTP Server Endpoint shall be able to serve the requested data bytes in a timely manner.
- The npt-pos and byte-pos of "0" shall be valid and shall point to the beginning of the content.

- Servers that receive a GET request that omits both Range and TimeSeekRange.dlna.org shall respond with content data from the absolute beginning (i.e. s_0) of the content.

A timely response is defined in 7.5.4.2.16.4.

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	[33]	4NTSX	

NOTE: This guideline clarifies the Mode=1 rules described in 7.5.4.2.16 MT "Limited Random Access Data Availability" Model, Requirement 7.5.4.2.16.4, by explaining how syntax tokens refer to the UCDAM data boundaries.

Essentially the byte range in the range-specifier indicate the data range for use in requests that use Range where the response will return in a timely manner.

Unlike Range, responses with TimeSeekRange.dlna.org are timely for the entire content binary.

Unlike Mode=0, a request that omits the Range and TimeSeekRange.dlna.org headers results with content data from the beginning of the content binary, which is required to be static and non-changing.

7.5.4.3.2.20.12

[Guideline] If mode-flag is '1', then the HTTP Server Endpoint shall be able to respond with the requested data bytes, even if the request specifies a bytes-range (in the HTTP request) that is not inclusively within the server's range-specifier (as indicated in the HTTP server's responses with availableSeekRange.dlna.org).

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	[33]	TSXZZ	

NOTE: The content data mapped by range-specifier is where timely responses are mandatory.

The content data for [s_0, s_N] is always available for transmission although the computational complexity to perform responses for the portions outside of the server's indicated range-specifier (in the availableSeekRange.dlna.org header value) can cause in responses that are not timely.

7.5.4.3.2.20.13

[Guideline] If mode-flag is '1' and the HTTP Server Endpoint responds with data bytes that is not inclusively within the server's range-specifier (as indicated in the HTTP server's responses with availableSeekRange.dlna.org), then the HTTP server, may begin its response with content data after 27 seconds.

[Attributes]

O	A	DMS +PU+	M-DMS	n/a	[33]	65V8D	

7.5.4.3.2.20.14

[Guideline] If an HTTP Server Endpoint supports the TimeSeekRange.dlna.org HTTP header for a content binary that operates under the "Limited Random Access Data Availability" model, it shall provide the npt-range in the availableSeekRange.dlna.org HTTP header field of the Target Response to an HTTP request with the getAvailableSeek.Range.dlna.org header HTTP header field.

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	[33]	ZZXWQ	

NOTE: The availableSeekRange.dlna.org and getAvailableSeekRange.dlna.org HTTP headers apply in scenarios where the HTTP server and content are governed by the "Limited Random Access Data Availability" model. Using these HTTP headers for other scenarios (such as those involving Full Random Access Data Availability model) is out-of-scope.

7.5.4.3.2.20.15

[GUIDELINE] If an HTTP Server Endpoint supports the Range HTTP header for a content binary that operates under the "Limited Random Access Data Availability" model, it shall provide the bytes-range in the availableSeekRange.dlna.org HTTP header field of the Target Response to an HTTP request with the getAvailableSeek.Range.dlna.org header HTTP header field.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	SXZZX	

7.5.4.3.2.20.16

[GUIDELINE] If the mode-flag=0, then an HTTP Client Endpoint shall not assume any meaning for an npt value 0 or a byte position of 0 returned in an availableSeekRange.dlna.org HTTP header.

(i.e. If mode-flag=0, then npt-pos=0 and byte-pos=0 have undefined behaviors under most situations. An HTTP Client Endpoint that specifies a byte-pos=0 or npt-pos=0 shall do so if the server's range-specifier includes 0 in the range.)

[ATTRIBUTES]

M	C	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33]	8DU64	A

NOTE: When the mode-flag is zero, then the "beginning of the content" becomes undefined. Clients are not to assume that npt or byte positions of 0 map to the beginning of the content. Furthermore, clients are not to assume that a GET request with a Range or TimeSeekRange.dlna.org that specifies 0 will succeed.

7.5.4.3.2.20.17

[GUIDELINE] If an HTTP Server Endpoint receives an HTTP GET request when s0-increasing is false and neither Range nor TimeSeekRange.dlna.org are supported by the HTTP Server Endpoint, then the HTTP Server Endpoint shall infer a byte-pos of 0 (i.e. respond from the beginning of the content binary).

[ATTRIBUTES]

M	A	DMS	M-DMS	MIU	[33]	5V8DU	

NOTE: This guideline ensures that HTTP clients will receive content in a manner consistent with the conventions established by HTTP.

7.5.4.3.2.21 MT HTTP Mapping for Byte-Based Seek and Time-Based Seek

7.5.4.3.2.21.1

[GUIDELINE] The Range HTTP header shall be used as the mechanism for byte-based seek operations on the HTTP transport protocol. The seek-able data range shall be equivalent to the random access data range of [r_0, r_N].

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[33]	BZ33H	

NOTE: Byte-based seek is a generic, transport layer term that applies only to Streaming transfers. The Range header is a specific transport layer mechanism for HTTP.

Note that the Range header is valid for Interactive and Background transfers, as described in 7.5.4.3.4.3 and 7.5.4.3.5.3. However the byte-based seek operation is only valid for Streaming transfers.

7.5.4.3.2.21.2

[GUIDELINE] The TimeSeekRange.dlna.org HTTP header shall be used as the mechanism for time-based seek operations on the HTTP transport protocol. The seek-able data range shall be equivalent to the random access data range of $[r_0, r_N]$.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[33]	XZZXW

NOTE: Time-based seek is a generic, transport layer term that applies only Streaming transfers. The TimeSeekRange.dlna.org header is a specific transport layer mechanism for HTTP.

Note that the time-based seek operation is valid only for Streaming transfers. Using the TimeSeekRange.dlna.org header is expressly prohibited for Background and Interactive transfers.

7.5.4.3.2.22 MT HTTP Header: Range (Server)

7.5.4.3.2.22.1

[GUIDELINE] If the Content Source indicates support for the Range HTTP header for a content binary (as defined in guidelines 7.4.1.3.19.1 and 7.4.1.3.28), then the HTTP Server Endpoint shall respond to Range HTTP requests for that content binary as defined in [33] with clarifications and constraints defined in guidelines 7.5.4.3.2.22.2 - 7.5.4.3.2.22.10.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	V8DU6

NOTE: HTTP server endpoints that receive an HTTP request with a Range header field are expected to respond in a specific manner. The rules below describe what an HTTP server needs if the URI can or cannot support requests with a specified Range.

7.5.4.3.2.22.2

[GUIDELINE] If an HTTP Server Endpoint receives the Range HTTP header in a HEAD request, then the HTTP server may respond without the Content-Range HTTP header.

[ATTRIBUTES]

O	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	33H53

NOTE: HTTP server endpoints are required to support the HEAD method. When an HTTP server gets a HEAD request with the Range option, the HTTP server can omit the Content-Range.

7.5.4.3.2.22.3

[GUIDELINE] The notation of Range header field for DLNA media transport shall be as defined in [33] with the restriction that only one interval is allowed.

The restricted syntax is as defined below:

- range-line = "Range" *LWS ":" *LWS range specifier
- range specifier = byte range specifier
- byte range specifier = bytes unit "=" byte range set
- bytes unit = "bytes"
- byte range set = byte range spec
- byte range spec = first byte pos "-" [last byte pos]

- first byte pos = 1*DIGIT
- last byte pos = 1*DIGIT

Note that the literal, "bytes", is case sensitive.

Examples:

- Range: bytes=1539686400-
- Range: bytes=1539686400-1540210688

[ATTRIBUTES]

M	L	DMS DMP DMR DMPr +PU+ +UP+ +DN+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33]	Z33H5

NOTE: This guideline simplifies the implementation of HTTP server endpoints by requiring only a subset of the allowed Range syntax afforded by [33]. Essentially DLNA implementations of HTTP clients can only assume that a DLNA implementation of an HTTP server will support Range values that indicate the first byte index and an optional last byte index. In summary, this restriction means that only a single range will be used within the Range header field.

While this guideline limits the number of range intervals to a single one, any other rules on syntax or semantics from [33] remain applicable.

7.5.4.3.2.22.4

[GUIDELINE] If an HTTP Server Endpoint returns data including the requested range (Target Response) with the HTTP response code 206 (Partial Content), then it shall specify the Content-Range header field.

Example of Content-Range header.

- Content-Range: bytes 1539686400-1540210688/9238118400

[ATTRIBUTES]

M	L	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	8WAAX

NOTE: These guidelines obligate an HTTP Server Endpoint to respond with a 206 response code (for a request that can be honored). These guidelines simplify HTTP Client Endpoint implementations as they limit the HTTP Server Endpoint to more predictable behavior.

Subclause 14.16 in [33] has examples on this guideline usage.

This guideline does not explain the relationship between first-byte-pos & last-byte-pos and the s_0 & s_N data boundaries. For example, if first-byte-pos=50 and last-byte-pos=200, then the response entity body will contain 151 bytes. The first byte of the entity-body will correspond to the 51st byte of the complete file and the last byte of the entity body will correspond to the 201st byte, relative to the complete content binary. This scenario does not at all imply that the 51st or 201st bytes (of the complete content binary) have any specific relationship to the s_0 or s_N data boundaries. Building on this example, if the s_N-increasing flag is true, then the complete binary is still increasing size. This results in a scenario where the 201st byte does not map to the s_N data boundary.

7.5.4.3.2.22.5

[GUIDELINE] If the HTTP Server Endpoint uses the Content-Range HTTP header, then the provided values shall be accurate with respect to the entity response body. Specifically,

- The value indicating the first-byte-pos shall properly match the first byte of the response entity body.

- The first-byte-pos in the response shall match the first-byte-pos in the request message.
- The value indicating the last-byte-pos shall properly match the last byte of the response entity body.
- The value indicating the instance-length shall indicate the length of the entire content binary or asterisk (*) if unknown.

Note that "accurate with respect to the entity response body" means that the guideline explains that the values used for the Content-Range header's first-byte-pos & last-byte-pos tokens need to correspond to the actual content data that is returned in the response entity body.

[ATTRIBUTES]

M	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	38J7V

7.5.4.3.2.22.6

[GUIDELINE] If the requested range is not valid for the resource with a URI specified in the HTTP request, the HTTP Server Endpoint shall respond with the HTTP response code of: 416 (Requested Range Not Satisfiable).

When encountering syntax errors with the Range HTTP header as defined in [33], the HTTP server shall respond using one of the following:

- Sending an HTTP response code 400 (Bad Request), or
- Ignoring the erroneous RANGE header

If the requested range is syntactically correct as defined in [33] but does not comply with guideline 7.5.4.3.2.22.3, the HTTP server shall respond using one of the following:

- Sending an HTTP response code 400 (Bad Request), or
- Ignoring the erroneous RANGE header, or
- Sending an HTTP response as defined in [33]

[ATTRIBUTES]

M	F	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	AAX28

NOTE: Ignoring the erroneous RANGE header means that the server will process the request as if such header were not present in the request.

RFC 2616 [33] subclause 14.35.1 defines what constitutes a valid range request and when the request is invalid. For example, for a file of size 500, a request for bytes 200 to 800 is valid. A request for bytes 700 to 800 is invalid.

7.5.4.3.2.22.7

[GUIDELINE] This guideline no longer applies.

[ATTRIBUTES]

						W38J7

7.5.4.3.2.22.8

[GUIDELINE] If an HTTP Server Endpoint can support HTTP requests with a specified Range for a particular URI, then the HTTP Server Endpoint should support persistent connections (HTTP/1.1) for that URI.

[ATTRIBUTES]

S	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	WAAX2

NOTE: Content that is requested with the Range option can often get many Range requests in a short period of time, potentially causing content serving devices to run out of available sockets.

7.5.4.3.2.22.9

[GUIDELINE] An HTTP Server Endpoint should accept and honor an HTTP/1.0 GET or HEAD requests for media transfers with the Range header field.

[ATTRIBUTES]

S	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	PW38J

NOTE: Although the Range option is not covered in the HTTP/1.0 specification, HTTP/1.0 clients can still benefit from having the ability to issue GET requests with a specified Range.

7.5.4.3.2.22.10

[GUIDELINE] This Guideline no longer applies.

[ATTRIBUTES]

						JS7HE

7.5.4.3.2.22.11

[GUIDELINE] If the Content Source does not indicate support for the Range HTTP header for a content binary (as defined in guidelines 7.4.1.3.19.1 and 7.4.1.3.28), then the HTTP Client Endpoint shall not issue HTTP GET and HEAD requests with the Range HTTP header.

[ATTRIBUTES]

M	L	DMP DMR +DN+ DMPr +DNSYNC+	M-DMP M-DMD	MIU	[33]	A57UJ

NOTE: This prohibition includes HTTP GET and HEAD requests with Range: bytes=0-.

7.5.4.3.2.22.12

[GUIDELINE] If the Range request is syntactically correct and the requested range is valid, and if the HTTP Server Endpoint returns data including the requested range (Target Response), then it shall respond to the HTTP GET request with the 206 (Partial Content) HTTP response code.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	7XZ95

NOTE: Comment in Requirement 7.5.4.3.2.22.4 also applies to this requirement.

7.5.4.3.2.23 HTTP Range Requirements for Printing System Usages

7.5.4.3.2.23.1

[GUIDELINE] The HTTP Server Endpoint of a +PR1+ shall implement the Range header by sending responses that include the Content-Range header for requested images. Returning 406 (Not Acceptable) is expressly prohibited.

[ATTRIBUTES]

M	A	+PR1+	n/a	n/a	[33]	8UZXT	

NOTE: DMPr devices will be able to print more efficiently when the HTTP server fully supports the Range header.

7.5.4.3.2.23.2

[GUIDELINE] The HTTP Client Endpoint of a DMPr may use the Range header to request portions of images when printing.

[ATTRIBUTES]

O	A	DMPr	n/a	n/a	[33]	C9LAC	

7.5.4.3.2.23.3

[GUIDELINE] The HTTP Server Endpoint of a Printing control point shall implement the Range header by sending responses that include the Content-Range header for requested XHTML documents. Returning 406 (Not Acceptable) is expressly prohibited.

[ATTRIBUTES]

M	A	+PR1+ +PR2+	n/a	n/a	[33]	SB3TF	

7.5.4.3.2.23.4

[GUIDELINE] The HTTP Client Endpoint of a DMPr shall be able to print images without the use of the Range header.

[ATTRIBUTES]

M	A	DMPr	n/a	n/a	[33]	V5NGA	

NOTE: DMS devices from previous guidelines are not required to support the Range header for image content.

7.5.4.3.2.24 MT HTTP Time-Based Seek (Server)

7.5.4.3.2.24.1

[GUIDELINE] If the Content Source indicates support for the TimeSeekRange.dlna.org HTTP header for a content binary (as defined in guidelines 7.4.1.3.19.1 and 7.4.1.3.28), then the HTTP Server Endpoint shall respond to the TimeSeekRange.dlna.org HTTP requests for that content binary with clarifications and constraints defined in guidelines 7.5.4.3.2.24.2 - 7.5.4.3.2.24.13.

[ATTRIBUTES]

M	A	DMS +PU+.	M-DMS	n/a	[33]	U8UZX	

NOTE: HTTP requests with the Range header field do not provide a very accurate experience when seeking to playback positions in variable bitrate encoded content. This HTTP header provides a way for DLNA HTTP clients to request an HTTP server to send the content bytes for a specified range of time.

Also note that HTTP clients can also use seek operations to implement a forward/backward variable speed playback capability.

7.5.4.3.2.24.2

[GUIDELINE] If an HTTP Server Endpoint receives the TimeSeekRange.dlna.org HTTP header in a HEAD request for streamed transfer, then it may respond without the TimeSeekRange.dlna.org HTTP header.

[ATTRIBUTES]

O	C	DMS +PU+	M-DMS	n/a	[33]	7ES2R	

NOTE: All HTTP server endpoints are required to support the HEAD method. When a streaming HTTP server endpoint gets a HEAD request with this option, the HTTP server can omit it from the response.

7.5.4.3.2.24.3

[GUIDELINE] The notation of the TimeSeekRange.dlna.org header field is defined as follows.

- TimeSeekRange-line = "TimeSeekRange.dlna.org" *LWS ":" *LWS range specifier
- range specifier = npt range [SP bytes-range]
- npt range = "npt" "=" npt-start- time "-" [npt-end- time] [instance-duration]
- npt-start-time = npt-time
- npt-end-time = npt-time
- instance-duration = "/" (npt-time | "*")
- npt time = <syntax as defined in 7.5.4.2.13 MT Normative Syntax for npt-time>
- bytes range = "bytes" "=" first byte pos "-" last byte pos instance-length
- first byte pos = 1*DIGIT
- last byte pos = 1*DIGIT
- instance-length = "/" (1*DIGIT | "*")

Note that literals, "npt" and "bytes", are case sensitive.

The npt-range specifies the range in normal playing time as found in subclause 3.6 of [27] with the exception of the concept of the "now" literal. It is used in the request and response.

The bytes-range specifies the range in bytes and it is used only in the response. Refer to 7.5.4.3.2.24.5.

The instance-duration specifies the duration of an entire content binary and is mandatory in the response and prohibited in the request. The asterisk "*" character means that the instance-duration is unknown at the time when the response was generated. Refer to 7.5.4.3.2.24.5 for more information.

The instance-length specifies the byte length of an entire content binary and bytes-range shall include instance-length in the response. The asterisk "*" character means that the instance-length is unknown at the time when the response was generated. Refer to 7.5.4.3.2.24.5 for more information.

Examples:

- TimeSeekRange.dlna.org : npt=335.11-336.08
- TimeSeekRange.dlna.org : npt=00:05:35.3-00:05: 37.5

Specifying the range value in the combination of npt-sec and npt-hhmmss, e.g. 335.11-00:05:37.5, is allowed, but not recommended.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+ +DN+ +DNSYNC+	M-DMS M-DMP M-DMD	MIU	[33] [27]	S2RRM	

NOTE: The field value specifies the requested range of a resource in absolute time. The range is specified by the start point and the endpoint. When both points are specified, the range value adopts the form:

TimeSeekRange.dlna.org: npt=time1-time2

The first value (time1) defines the start time, while the second value (time 2) defines the end time.

In forward scan mode (positive play speed values): time1 <= time2 (as defined in 7.5.4.3.2.24.12)

In backward scan mode (negative play speed values): time1 >= time2 (as defined in 7.5.4.3.2.24.12)

The end time can be omitted resulting in a expression of the form:

TimeSeekRange.dlna.org: npt=time1-

If the end time is omitted in forward scan mode, it means the end of the resource is specified. If the end time is omitted in backward scan mode, it means the beginning of the resource is specified.

7.5.4.3.2.24.4

[GUIDELINE] If an HTTP Server Endpoint returns data in forward scan mode including the requested time range (Target Response), then it shall return the content bytes for a time-range that starts at or before the requested start time and ends at or after the requested end time.

If an HTTP Server Endpoint returns data in backward scan mode including the requested time-range (Target Response), then it shall return the content bytes for a time-range that starts at or after the requested start time and ends at or before the requested end time.

The following exceptions are allowed for the “Full Random Access Data Availability” model:

- In forward scan mode an HTTP Server Endpoint may ignore the requested end time and return the range data up to the end of the content data. In backward scan mode an HTTP Server Endpoint may ignore the requested end time and return the range data up to the beginning of the content data.

The following exception is allowed for both the "Full Random Access Data Availability" model and the "Limited Random Access Data Availability" model:

- HTTP Server Endpoint may round up or down to one decimal place, the npt time values specified in the HTTP response TimeSeekRange.dlna.org, compared to the actual returned data in the response body.

Examples

- TimeSeekRange.dlna.org : npt=335.1-336.1/40445.4
- TimeSeekRange.dlna.org : npt=00:05:35.2-00:05:38.1/*

In forward scan mode, if the requesting endpoint specifies an end time beyond the end of the content data, then the HTTP Server Endpoint shall return the range data to the end of the content data.

In backward scan mode, if the requesting endpoint specifies a start time beyond the end of the content data, then the HTTP Server Endpoint shall return the range data to the end of the content data.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	57UJY

NOTE: This guideline obligates an HTTP server to respond with a time range (of bytes) that is close to the time range specified in the request.

7.5.4.3.2.24.5

[GUIDELINE] If an HTTP Server Endpoint supports both byte-based seek and time-based-seek operations for a resource, then it shall specify a byte-range value as well as an npt-range value in the HTTP Target Response to the HTTP request with the TimeSeekRange.dlna.org header field, unless the following exceptions apply.

Exceptions:

- If all of the following conditions are true, then the npt-range shall be included (i.e. server is permitted to omit the bytes-range token in the TimeSeekRange.dlna.org, if the data range is unknown).
- The data access model is the "Limited Random Access Data Availability" model.
- The mode-flag (as indicated in availableSeekRange.dlna.org) is "1".
- The request omitted the Range header and only specified TimeSeekRange.dlna.org. (Note that the Range header always takes precedence over the TimeSeekRange.dlna.org, as indicated in 7.5.4.3.3.17.) If the request incudes the PlaySpeed.dlna.org header with positive or negative speed values, then the response is permitted to omit the bytes-range token in the TimeSeekRange.dlna.org header.

When specified, the npt-range and byte-range shall include instance-duration and instance-length respectively.

Examples:

- TimeSeekRange.dlna.org : npt=335.1-336.1/40445.4 bytes=1539686400-1540210688/304857907200
- TimeSeekRange.dlna.org : npt=00:05:35.2-00:05:38.1/* bytes=1539686400-1540210688/*

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	7UJYL

NOTE: The time-based-seek capability is useful for seeking to playback positions in variable bit rate encoded content; but it is not useful to retrieve subsequent data blocks. For Trick Mode playback after an initial time-based-seek, use of multiple HTTP GET requests with the Range header field to retrieve subsequent content data is encouraged. To support this functionality, byte range value is also specified in addition to time range value for the response to a Time-Based-Seek only when byte-seek is also supported for the resource.

This functionality requires an HTTP server (that supports both TimeSeekRange.dlna.org and Range) to specify both the time-range and a byte-range in the HTTP response's headers.

7.5.4.3.2.24.6

[GUIDELINE] If an HTTP Server Endpoint returns data (Target Response) from a GET request with time range, then the response entity body data should start at a decoder friendly point (for example the start of the GOP in a video sequence).

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	[33]	ES2RR

NOTE: This guideline recommends that stream segments returned by servers start, if possible, with a recognizable decoding entry point, such as the headers in a group of pictures (GOP) in MPEG-2.

7.5.4.3.2.24.7

[GUIDELINE] If an HTTP Server Endpoint returns data including the requested time range (Target Response) with the HTTP response code 200 (OK), then it shall specify the TimeSeekRange.dlna.org header field to indicate the time range of the content data that is returned in the HTTP response.

The npt-range shall include the instance-duration.

Examples

- TimeSeekRange.dlna.org : npt=335.10-336.10/40445.4
- TimeSeekRange.dlna.org : npt=00:05:35.3-00:05: 37.5/*
- For a response in backward scan mode: TimeSeekRange.dlna.org : npt=897.5-241.5/3966.3

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	YC9LA	

NOTE: These guidelines obligate an HTTP server to respond with either a 200 response code (for a time range request that can be honored) or a 416 response code (for time range requests that cannot be honored). The guideline simplifies HTTP client implementations as it makes limits the behavior of the HTTP server more predictable.

7.5.4.3.2.24.8

[GUIDELINE] If the requested time range is not valid for the resource with URI specified in the HTTP GET request, then the HTTP streaming server shall respond with the HTTP response error code of: 416 (Requested Range Not Satisfiable).

Interpretation of not valid includes the following types of errors:

- The requested time range is not within the time boundaries of the actual content.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	9LACX	

NOTE: The HTTP server indicates support for the TimeSeekRange.dlna.org header in the DLNA.ORG_OP parameter in guideline 7.4.1.3.19.1.

7.5.4.3.2.24.9

[GUIDELINE] If the requested time range is not syntactically correct, then the HTTP streaming server shall return the HTTP response error code of 400 (Bad Request).

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	YV45Z	

7.5.4.3.2.24.10

[GUIDELINE] This Guideline no longer applies.

[ATTRIBUTES]

						8T5GV	

7.5.4.3.2.24.11

[GUIDELINE] If the Content Source does not indicate support for the TimeSeekRange.dlna.org HTTP header for a content binary (as defined in guidelines 7.4.1.3.19.1 and 7.4.1.3.28), then the HTTP Client Endpoint shall not issue HTTP GET and HEAD requests with the TimeSeekRange.dlna.org HTTP header.

[ATTRIBUTES]

M	L	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33]	8V5NG

7.5.4.3.2.24.12

[GUIDELINE] In forward scan mode (positive play speed values) the start time in the range defined by the TimeSeekRange.dlna.org header shall be smaller than or equal to the end time.

In backward scan mode (negative play speed values) the start time in the range defined by the TimeSeekRange.dlna.org header shall be larger than or equal to the end time.

Non-compliance with these requirements results in a syntax error in the TimeSeekRange.dlna.org header. See guideline 7.5.4.3.2.24.9 for behavior in the presence of a syntactically wrong TimeSeekRange.dlna.org header.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+ +DN+ +DNSYNC+	M-DMS M-DMP M-DMD	MIU	n/a	**ENIDK**

7.5.4.3.2.24.13

[GUIDELINE] If the TimeSeekRange.dlna.org request is syntactically correct and the requested time range is valid, and if the HTTP Server Endpoint returns data including the requested time range (Target Response) then it shall respond to the HTTP GET request with the 200 (OK) HTTP response code.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	WS9TZ

7.5.4.3.2.25 MT HTTP Chunked Transfer Coding

7.5.4.3.2.25.1

[GUIDELINE] HTTP Servers Endpoints may use *Chunked Transfer Coding* in response to HTTP/1.1 GET requests.

[ATTRIBUTES]

O	R	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	58V5N

NOTE: Chunked Transfer Coding is an HTTP response encoding methodology that can only be used in response to HTTP/1.1 requests by HTTP/1.1 servers.

7.5.4.3.2.25.2

[GUIDELINE] HTTP Server Endpoints shall not use *Chunked Transfer Coding* in response to HTTP/1.0 GET requests.

[ATTRIBUTES]

M	R	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	3KYV4

7.5.4.3.2.25.3

[GUIDELINE] HTTP Client Endpoints may use *Chunked Transfer Coding* in HTTP/1.1 POST requests.

[ATTRIBUTES]

O	R	+UP+ +UPSYNC+	M-DMU	MIU	[33]	XJSB3

7.5.4.3.2.26 MT Baseline Transport: HTTP Client Endpoints

7.5.4.3.2.26.1

[GUIDELINE] HTTP Client Endpoints used for media transport purposes shall implement HTTP/1.0, HTTP/1.1, or both.

[ATTRIBUTES]

M	L	DMP DMR DMPr +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33]	KYV45

NOTE: HTTP client endpoints are restricted to HTTP versions that HTTP server endpoints are prepared to support.

7.5.4.3.2.26.2

[GUIDELINE] HTTP Client Endpoints shall not report a higher version of HTTP than is actually supported by the implementation.

[ATTRIBUTES]

M	R	DMP DMR DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	[33]	TF57C

NOTE: For example an HTTP client endpoint that does not support *Chunked Transfer Coding* responses will never issue an HTTP/1.1 GET request.

7.5.4.3.2.26.3

[GUIDELINE] HTTP/1.1 Client endpoints shall be able to process HTTP/1.1 *Chunked Transfer Coding* responses.

[ATTRIBUTES]

M	R	DMP DMR DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	[33]	GAA77

NOTE: When making HTTP/1.1 requests, it is important that HTTP client endpoints properly handle responses encoded with *Chunked Transfer Coding*.

7.5.4.3.2.26.4

[GUIDELINE] HTTP/1.1 Client Endpoints shall be prepared to properly handle an HTTP response code of 1xx from HTTP servers, even when not expected.

[ATTRIBUTES]

M	R	DMP DMR DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	[33]	3TF57

NOTE: A 1xx can be generated by the server regardless of whether or not the client issued an HTTP request encoded with *Chunked Transfer Coding*.

See subclause 10.1 of [33] for more information.

7.5.4.3.2.27 MT HTTP Header: Range (Client)

7.5.4.3.2.27.1

[GUIDELINE] HTTP Client Endpoints shall not use multiple range specifiers nor use suffix byte range spec (as defined in [33]) in HTTP requests.

[ATTRIBUTES]

M	L	DMP DMR DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	[33]	B3TF5	

NOTE: This guideline simplifies the implementation of HTTP servers by not requiring support of multiple ranges or suffix byte range spec.

7.5.4.3.2.28 MT HTTP Persistent Connection Usage for Clients

7.5.4.3.2.28.1

[GUIDELINE] HTTP/1.1 Client Endpoints should use HTTP/1.1 persistent connections.

[ATTRIBUTES]

S	R	DMP DMR DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	[33]	45Z8H	

NOTE: Implementing this guideline reduces the setup/teardown load on HTTP Server Endpoints. Furthermore, HTTP Server Endpoints will be able to reserve the allocated socket for the requesting client. Clients that do not use HTTP/1.1 persistent connections can encounter a scenario where an HTTP Server Endpoint does not answer the subsequent requests because it has run out of sockets.

7.5.4.3.2.28.2

[GUIDELINE] This guideline no longer applies.

[ATTRIBUTES]

						5Z8H8	D

7.5.4.3.2.29 MT HTTP Inactivity Timeout

7.5.4.3.2.29.1

[GUIDELINE] HTTP Client Endpoints should close persistent (HTTP/1.1) connections after completing all outstanding HTTP transactions and within 30 seconds of inactivity has passed.

[ATTRIBUTES]

S	A	DMP DMR DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	[33]	V45Z8	

NOTE: This ensures that sockets do not remain consumed after a content transfer has successfully completed.

7.5.4.3.2.29.2

[Guideline] If an HTTP server detects 5 minutes of inactivity after a POST transaction, then the HTTP server should close persistent (HTTP/1.1) connections.

[Attributes]

S	A	DMS	M-DMS	n/a	[33]	NGAA7	

NOTE: This would ensures that sockets do not remain consumed by an Upload Controller.

7.5.4.3.2.30 MT HTTP Header Parsing (Client)

[Guideline] HTTP Client Endpoints shall gracefully skip over unsupported HTTP header fields. Under no circumstances can an HTTP client fail to process a properly formatted HTTP response because of an unrecognized or unsupported HTTP header field.

[Attributes]

M	R	DMP DMR DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	[33]	5NGAA	

NOTE: Incorrect HTTP header parsing has been the source of numerous compatibility issues during plugfest events.

7.5.4.3.2.31 MT HTTP Maximum Header Size

[Guideline] HTTP Client and Server Endpoints shall use a total HTTP header size that is less than or equal to 8192 bytes (8 KB) when sending an HTTP request or HTTP response.

The total HTTP header size is the total number of bytes from the first byte in the start-line token and the last byte of the CRLF token, as used in the generic-message token defined in subclause 4.1 of [33], as quoted in the syntax below.

- generic-message = start-line *(message-header CRLF) CRLF [message-body]

[Attributes]

M	L	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33]	CX7UF	

NOTE: This provides a reasonable assumption as to how much memory is necessary is for all HTTP headers in an HTTP request or response.

If determined to be necessary, future DLNA guidelines can define a parameter in the USER-AGENT to indicate a DLNA version for governing total HTTP header sizes.

7.5.4.3.2.32 MT HTTP Status Code Precedence

[Guideline] If a DLNA guideline specifies an HTTP status code that involves a DLNA defined HTTP header, then the DLNA guideline's HTTP status code shall take precedence over those specified in [33].

[Attributes]

M	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	ACX7U	

NOTE: As a general rule, HTTP status codes for DLNA defined HTTP headers take precedence over HTTP status codes. For standard HTTP headers, [33] status codes apply. For HTTP responses that involve DLNA defined HTTP headers, the DLNA specified status code applies when there exists an ambiguity between using a DLNA-specified error code and an error code made by [33]. Whenever possible, DLNA guidelines align with [33].

7.5.4.3.2.33 MT Transfer Mode Indication

7.5.4.3.2.33.1

[Guideline] The syntax for the transferMode.dlna.org HTTP header value is as follows.

- transferMode-line= "transferMode.dlna.org" *LWS ":" *LWS mode
- mode = "Streaming" | "Interactive" | "Background"

A value of "Background" for an HTTP request indicates that the HTTP client would like the transfer of content to be a Background Transfer.

A value of "Interactive" for an HTTP request indicates that the HTTP client is requesting an Interactive Transfer.

A value of "Streaming" for an HTTP request indicates that the HTTP client wishes a Streaming Transfer of the content.

A value of "Background" for an HTTP response indicates that the HTTP server is attempting a Background Transfer of the content.

A value of "Interactive" for an HTTP response indicates that the HTTP server is attempting an Interactive Transfer of the content.

A value of "Streaming" for an HTTP response indicates that the HTTP server is attempting a Streaming Transfer of the content.

Note that the mode token values ("Streaming", "Interactive", and "Background") are case sensitive values.

Example

- transferMode.dlna.org: Interactive

[Attributes]

M	A	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33]	LACX7

NOTE: This header is carries the Transfer Mode as listed in the introduction.

7.5.4.3.2.33.2

[Guideline] An HTTP server that receives an HTTP request from an HTTP client of a DLNA endpoint that omits the transferMode.dlna.org header shall treat it as equivalent to the header-value pair of transferMode.dlna.org:Streaming for content binaries of the AV or Audio media classes or transferMode.dlna.org:Interactive for all other content binaries.

[Attributes]

M	A	DMS	M-DMS	n/a	[33]	RRM6T

NOTE: HTTP transfers that do not include this header are treated as Streaming Transfers for AV or Audio media classes and as Interactive Transfers for all other content binaries.

7.5.4.3.2.33.3

[GUIDELINE] The transferMode.dlna.org header may be used by HTTP clients and servers in requests and responses. The transferMode.dlna.org header is used to convey the transfer mode information as specified in guideline 7.5.4.2.3.

[ATTRIBUTES]

O	C	DMS DMP DMR DMPr +PU+ +DN+ +PR1+ +PR2+ +UP+ +UPSYNC+ +DNSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33]	YL6RN

NOTE: This guideline clarifies the use of the transferMode.dlna.org, according to guideline 7.5.4.2.3.

7.5.4.3.2.33.4

[GUIDELINE] If an HTTP Client Endpoint requests a transfer mode that is not valid for the media class of data that is being exchanged or is not supported by an HTTP Server Endpoint for the given content binary, the HTTP Server Endpoint shall respond with error code 406 (Not Acceptable).

[ATTRIBUTES]

M	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	2RRM6

NOTE: The list of transfer modes for the given media class is given in guideline 7.5.4.2.3.2. However, a server might restrict the list further if the content warrants it. For example, for live captured content, Background Transfer might not be available. For details on the Media Transfer Modes that are available for a particular content binary within a DMS structure, see guideline 7.4.1.3.23.2.

7.5.4.3.2.33.5

[GUIDELINE] All DLNA endpoints communicating with a v1.0 DMS shall tolerate HTTP responses that omit the transferMode.dlna.org HTTP header.

Tolerate means that the HTTP client shall "parse and ignore" or "parse and interpret" the HTTP response.

(i.e. All DLNA endpoints communicating with a v1.0 DMS shall assume that the media will be returned without the transferMode.dlna.org HTTP header and using a best effort QoS level.)

[ATTRIBUTES]

M	C	DMP DMR DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMP M-DMD M-DMU	MIU	[33]	RM6TQ

NOTE: All v1.0 DMS media transfers were assumed to be for immediate rendering and at a default best effort QoS level. A v1.5 entity communicating with a v1.0 DMS needs to allow any media transfer to have these parameters.

7.5.4.3.2.34 MT Caching Directives for HTTP 1.0

7.5.4.3.2.34.1

[GUIDELINE] If applicable-http-endpoints want to modify the default caching behavior of intermediate HTTP caches, they shall do so in a manner compliant with [21].

Applicable-http-endpoints specifically refers to HTTP Server and Client Endpoints participating in transfer operations of Cacheable Content using:

- HTTP/1.0, and
- GET requests and responses.

This definition of applicable-http-endpoints is valid only for this guideline.

[ATTRIBUTES]

M	C	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[21]	JYL6R

NOTE: Per HTTP/1.0 specifications, intermediate caching operations are allowed by default when content is transferred using the GET method.

7.5.4.3.2.34.2

[GUIDELINE] If an HTTP Server Endpoint transfers Non-Cacheable Content using HTTP/1.0 GET responses, then the HTTP Server Endpoint shall prevent intermediate caching by including this directive amongst the HTTP headers in the response.

- Pragma: no-cache

[ATTRIBUTES]

M	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[21]	J7V9I

NOTE: Non-cacheable content includes the following.

- live content and other forms of broadcast streams
- content data that includes the TimeSeekRange.dlna.org or PlaySpeed.dlna.org HTTP headers
- content binaries that are dynamically generated in such a way that the content binary streams can differ on 2 different transactions (e.g. smart transcoding engines that perform transrating based on network throughput)

7.5.4.3.2.34.3

[GUIDELINE] If HTTP Server Endpoints transfer Audio or AV content binaries and the response includes one or both of listed HTTP headers, then the responses shall prevent caching, per guidelines 7.5.4.3.2.35.1 and 7.5.4.3.2.35.2.

- TimeSeekRange.dlna.org
- PlaySpeed.dlna.org

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	n/a	UJYL6

NOTE: In HTTP 1.0 no method exists to signal the different bit stream variations that result from the use of Time Seek and Play Speed operations. For this reason, intermediate caching operations have to be avoided.

7.5.4.3.2.35 MT Caching Directives for HTTP 1.1

7.5.4.3.2.35.1

[GUIDELINE] If applicable-http-endpoints want to modify the default caching behavior of intermediate HTTP caches, they shall do so in a manner compliant with [33].

Applicable-http-endpoints specifically refers to HTTP Server and Client Endpoints participating in transfer operations of Cacheable Content using the following:

- HTTP/1.1, and
- GET requests and responses.

This definition of applicable-http-endpoints is valid only for this guideline.

[ATTRIBUTES]

M	C	DMS DMP DMR DMPr +PU+ +DN+ +UP+ +PR1+ +PR2+ +DNSYNC+ +UPSYNC+	M-DMS M-DMP M-DMD M-DMU	MIU	[33]	XTY7C	

NOTE: Per HTTP/1.1 specifications, intermediate caching operations are allowed by default when content is transferred using the GET method.

7.5.4.3.2.35.2

[GUIDELINE] HTTP Server Endpoints that transfer Non-Cacheable Content using HTTP/1.1 GET responses shall prevent intermediate caching by including among the HTTP response headers both of the following directives:.

- Pragma: no-cache
- Cache-control: no-cache

[ATTRIBUTES]

M	C	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	ZXTY7	

7.5.4.3.2.35.3

[GUIDELINE] If an HTTP Server Endpoint transfers Audio or AV content binaries using HTTP/1.1 GET responses that include one or both of these HTTP headers, then such transfers should be marked as cacheable.

- TimeSeekRange.dlna.org
- PlaySpeed.dlna.org

[ATTRIBUTES]

S	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	n/a	X28MG	

NOTE: Unlike HTTP/1.0, the newer version HTTP/1.1 includes caching directives that, when used, enable transparent caching even when the transferred objects include headers that modify the object's binary representation. In DLNA, Time Seek and Play Speed headers modify the binary representation of the object that is transferred using the HTTP protocol. The "Vary" header described below can be used to restore transparent caching for these objects.

7.5.4.3.2.35.4

[GUIDELINE] If an HTTP Server Endpoint transfers an Audio or AV content binaries that permits variable play speed and time-based seek operations for cacheable content transported in an HTTP/1.1 GET response, then the HTTP Server Endpoint shall include a "Vary" HTTP header as defined in [33].

The "Vary" header shall list either or both of the following two arguments to inform caches of the corresponding supported operations:

- TimeSeekRange.dlna.org
- PlaySpeed.dlna.org

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	UZXTY

NOTE: The Vary header serves to indicate potential intermediate caches that certain headers create variants of the object binaries defined by its URI. For example, a 30-minute media stream requested in full is a different object variant when only the first 3 minutes of the stream are requested using Time Seek operations.

Note: Per [33] the Vary header has to be included whenever the server responds to a client request for the CDS object. The Vary header has to be included even if the request does not include Time Seek or Play Speed headers.

7.5.4.3.2.35.5

[GUIDELINE] If applicable-http-endpoints want to supersede the default caching prohibition for POST transfers, they shall do so in a manner compliant with [33].

Applicable-http-endpoints specifically refers to HTTP Server and Client Endpoints participating in upload transaction using

- HTTP/1.1 and
- POST requests and responses.

[ATTRIBUTES]

M	C	DMS +UP+ +UPSYNC+	M-DMS M-DMU	MIU	[33]	8J7V9

NOTE: Per HTTP/1.1 specifications, intermediate caching operations are prohibited by default when content is transferred using the POST method.

7.5.4.3.2.36 MT/BCM HTTP Header:peerManager.dlna.org

7.5.4.3.2.36.1

[GUIDELINE] HTTP Client Endpoints may use the peerManager.dlna.org HTTP header to communicate the identity of the UPnP AV ConnectionManager service to the Content Source Endpoint in HTTP GET requests.

[ATTRIBUTES]

O	A	DMR	n/a	n/a	[33]	AX28M

NOTE: The peerManager.dlna.org HTTP Header is not required. However, this feature is useful for HTTP Clients that want to facilitate BCM on DMS with BCM support.

7.5.4.3.2.36.2

[GUIDELINE] The value of the peerManager.dlna.org HTTP header shall be the same value as the PeerConnectionManager, as described in UPnP AV Architecture.

The notation of the peerManager.dlna.org HTTP header field is defined as follows:

- peerManager-line = "peerManager.dlna.org" *LWS ":" *LWS peer-connection-manager
- peer-connection-manager = udn-token "/" serviceId-token
- udn-token = <case-insensitive UDN of the UPnP AV MediaRenderer device>

- serviceId-token = <case-sensitive ServiceID of the ConnectionManager service, associated with the MediaRenderer device identified by udn-token>

Example

- peerManager.dlna.org : uuid:12345678123456781234567812345678/urn:schemas-upnp-org:serviceId:ConnectionManager

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[33]	7V9I4	

7.5.4.3.2.37 MT/BCM HTTP Header:friendlyName.dlna.org

7.5.4.3.2.37.1

[GUIDELINE] HTTP Client Endpoints may use the friendlyName.dlna.org HTTP header to communicate a user friendly name to the Content Source Endpoint in HTTP GET requests.

[ATTRIBUTES]

O	A	DMP +UP+ +DN+ +DNSYNC+ +UPSYNC+	n/a	MIU	[33]	H53W5	

NOTE: The friendlyName.dlna.org HTTP Header is not required. However, this feature is useful for HTTP Clients that want to facilitate BCM on DMS with BCM support.

7.5.4.3.2.37.2

[GUIDELINE] The notation of the friendlyName.dlna.org header field is defined as follows:

- friendlyName-line = "friendlyName.dlna.org" *LWS ":" *LWS "friendly-name-token
- friendlyName-token = <case sensitive string, limited to 128 bytes in its UTF-8 encoded form>

[ATTRIBUTES]

M	A	DMP +UP+ +DN+ +DNSYNC+ +UPSYNC+	n/a	MIU	[33]	28MG8	

7.5.4.3.2.38 MT/BCM scid.dlna.org HTTP header

7.5.4.3.2.38.1

[GUIDELINE] If a UPnP AV MediaServer supports BCM, then it shall use the scid.dlna.org HTTP header in HTTP responses to identify the ConnectionID associated with the underlying transport layer connection.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[33]	3H53W	

7.5.4.3.2.38.2

[GUIDELINE] The syntax of the scid.dlna.org HTTP header shall be as follows:

- scidheader = "scid.dlna.org" *LWS ":" *LWS connection-id
- connection-id = <the ConnectionID value associated with the underlying transport connection>

[ATTRIBUTES]

M	A	DMS DMP DMR +UP+ +DN+ +DNSYNC+ +UPSYNC+	M-DMS	MIU	[33]	ZXWQG

7.5.4.3.2.38.3

[GUIDELINE] HTTP Client Endpoints may use scid.dlna.org in subsequent HTTP requests to identify a known and associated UPnP AV Connection only if the HTTP requests are for the same URI.

[ATTRIBUTES]

O	A	DMP DMR +UP+ +DN+ +DNSYNC+ +UPSYNC+	M-DMP	MIU	[33]	53W5Q

NOTE: This guideline allows Content Receivers to provide the ConnectionID in the transport layer requests. This guideline assumes the HTTP client of the Content Receiver acquired the ConnectionID in an earlier HTTP response. This mechanism can be used to combine logically related but separated transport layer requests for the same content URI.

7.5.4.3.3 HTTP Transport: Streaming Transfer Guidelines

7.5.4.3.3.1 MT streaming transferMode.dlna.org HTTP Header

7.5.4.3.3.1.1

[GUIDELINE] If an HTTP Client Endpoint requests a streaming transfer of data, then it shall specify the transferMode.dlna.org HTTP header and the header shall have a value of "Streaming".

[ATTRIBUTES]

M	A	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33]	DU649

NOTE: Version 1.0 clients will still request streaming transfers without using the transferMode.dlna.org HTTP header. Guideline 7.5.4.3.2.33.2 requires that servers that receive such a request treat it as equivalent to a request for a streaming transfer for Audio and AV media classes

7.5.4.3.3.1.2

[GUIDELINE] Unless a streaming HTTP Server Endpoint responds with an error response code, it shall respond to HTTP HEAD or GET requests for a streaming transfer (as defined in 7.5.4.3.3.1.1) by sending a response that contains the transferMode.dlna.org HTTP header and gives it a value of "Streaming".

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	6498V

7.5.4.3.3.2 MT HTTP Play Media Operation

[GUIDELINE] A streaming HTTP Client Endpoint shall use the HTTP GET method when using the HTTP transport protocol for initiating the Play media operation.

[ATTRIBUTES]

M	L	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33]	U6498

NOTE: This guideline specifies the normative way to request content.

7.5.4.3.3.3 MT HTTP Stop Media Operation

7.5.4.3.3.3.1

[GUIDELINE] A streaming HTTP Client Endpoint should implement the Stop media operation by disconnecting the TCP connection of the HTTP transaction.

[ATTRIBUTES]

S	A	+DN+ +DNSYNC+	M-DMD	MIU	[33]	XWQGT	C

NOTE: This guideline recommends the way to stop a media stream. Although HTTP clients can technically stall the TCP connection, that technique is discouraged. The preferred technique makes better use of an HTTP Server Endpoint's platform resources, which can be limited.

Note that HTTP Client endpoints are required to visually and/or aurally stop rendering of content, although the continuation of data streaming from the HTTP Server Endpoint is permissible for data buffering/caching purposes.

7.5.4.3.3.3.2

[GUIDELINE] An audio or AV streaming HTTP Client Endpoint shall implement the Stop media operation by disconnecting the TCP connection of the HTTP transaction.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[33]	CWCV2	N

NOTE: This technique makes better use of an HTTP Server Enpoint's platform resources, which can be limited.

Note that HTTP Client endpoints are required to visually and/or aurally stop rendering of content, although the continuation of data streaming from the HTTP Server Endpoint is permissible for data buffering/caching purposes.

7.5.4.3.3.4 MT HTTP Pause/Pause-Release Media Operation

7.5.4.3.3.4.1

[GUIDELINE] A streaming HTTP Client Endpoint should implement the Pause media operation by the following method:

- Connection Stalling: Suspending the reading of data from the HTTP connection. Resuming the reading of data from the HTTP connection is the mechanism for Pause-Release.

[ATTRIBUTES]

S	L	DMP DMR	M-DMP	MIU	[33]	WQGTQ	C

7.5.4.3.3.4.2

[GUIDELINE] If a streaming HTTP Client Endpoint performs a Pause media operation, then the HTTP Client Endpoint shall support all of the following Pause media operation methods:

- Disconnecting and Seeking: Disconnecting the HTTP connection and using byte-base seek transport layer features as the mechanism for Pause-Release.
- Disconnecting and Seeking: Disconnecting the HTTP connection and using time-based seek transport layer features as the mechanism for Pause-Release.

[ATTRIBUTES]

M	L	DMP DMR	M-DMP	MIU	[33]	CZ794	N

NOTE: This guideline is applicable to audio Media Class and AV Media Class.

7.5.4.3.3.4.3

[GUIDELINE] If a streaming HTTP Client Endpoint wants to pause the current media stream, it shall first ensure that all of the necessary media operations and information are available to resume the play (Pause-Release) of the media stream.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[33]	498VR	

NOTE: The ability to do a Pause-Release media operation depends on both the Content Receiver and the Content Source sharing support for at least one of the following HTTP transport features: byte-based seek, time-based seek, or Connection Stalling.

7.5.4.3.3.5 MT HTTP Pause/Pause-Release Media Operation: Disconnecting and Seeking Method

7.5.4.3.3.5.1

[GUIDELINE] If a streaming HTTP Client Endpoint performs a Pause media operation using the Disconnecting and Seeking method, it shall support the Pause-Release operation by using byte-based seek or time-based seek transport layer features. Before using this form of Pause/Pause-Release mechanism, the streaming HTTP Client Endpoint shall verify that the Content Source supports the necessary transport layer feature.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[33]	3W5QP	

NOTE: The HTTP connection can be closed for different reasons. If the connection is disconnected, the streaming HTTP Client Endpoint can use a Seek media operation (see 7.5.4.3.3.7.1) to Pause-Release the playback.

7.5.4.3.3.5.2

[GUIDELINE] If a streaming HTTP Client Endpoint supports the Pause media operation using the Disconnecting and Seeking method, it should support the byte-based seek transport layer feature.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[33]	98VRE	

NOTE: This guideline recommends that a Content Source supports byte-based seek, as defined in 7.5.4.3.2.22 MT HTTP Header: Range (Server).

7.5.4.3.3.5.3

[GUIDELINE] If a streaming HTTP Client Endpoint supports the Pause media operation using the Disconnecting and Seeking method, it may perform the Pause media operation by first suspending the reading of data from the HTTP connection (as described for the Connection Stall method). When the streaming HTTP Client Endpoint detects a TCP-layer disconnect, it may perform the Pause-Release media operation using the time-based seek or byte-based seek transport layer feature that is supported by the Content Source.

[ATTRIBUTES]

O	A	DMP DMR	M-DMP	MIU	[33]	W5QPO	

NOTE: Using TCP flow control to stall/pause the flow of data can enable quick Pause-Release behavior. Content Sources are recommended to support the Connection Stalling method, in addition to byte-based seek or time-based seek transport layer features.

7.5.4.3.3.6 MT HTTP Pause/Pause-Release Media Operation: Connection Stalling Method

7.5.4.3.3.6.1

[GUIDELINE] If a streaming HTTP Client Endpoint performs a Pause media operation using the Connection Stalling method it shall verify the http-stalling parameter in the 4th field of the res@protocolInfo is present and set to true for a content binary.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[33]	8MG85

NOTE: A Pause media operation initiated by the Connection Stalling method does not require the use of time-based seek or byte-based seek since the Content Source maintains the TCP connection, using standard TCP flow control. Content Sources can choose to support only the Connection Stalling method for some content binaries, such as those created by dynamic, real-time transcoding.

7.5.4.3.3.6.2

[GUIDELINE] When the HTTP connection is lost for any reason, the streaming HTTP Client Endpoint may attempt to use time-based seek or byte-based seek transport layer features to Pause-Release the media stream.

[ATTRIBUTES]

O	A	DMP DMR	M-DMP	MIU	[33]	MG857

NOTE: This guidelines assumes both the Content Receiver and Content Source support the same transport layer features.

7.5.4.3.3.6.3

[GUIDELINE] If the http-stalling flag is true for a content binary, then the streaming HTTP Server Endpoint shall allow Connection Stalling for an indefinite amount of time on that content binary.

Equivalently, streaming HTTP Server Endpoints that support the Connection Stalling method for a content binary shall be able to maintain the HTTP connection and shall not use an HTTP connection inactivity timeout to terminate the HTTP connection.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	V9I4U

NOTE: This guideline prohibits using an HTTP-inactivity timeout to terminate HTTP connections that are being paused through Connection Stalling.

The guideline permits the streaming HTTP Server Endpoint to terminate HTTP connections for the following scenarios:

- when the Content Receiver terminates the HTTP connection,
- the underlying TCP transport session is broken or disconnected,
- system events on the streaming HTTP Server Endpoint: user-initiated termination of streams, scheduled recording events, configurable policies for idle connections, etc.

Although this guideline requires a streaming HTTP Server Endpoint to allow Connection Stalling for an indefinite period of time, a streaming HTTP Server Endpoint can provide users with the ability to terminate the HTTP connections. Many details in this area are out of scope, but this guideline accounts for the following types of possibilities:

- A local UI, associated with the Content Source, allows the user to manually terminate the HTTP connections.

- The Content Source has user-configurable policies that can override the default behavior of indefinite connection stalling by terminating HTTP connections that have been inactive for a lengthy time. These guidelines do not define a minimum time but the suggested minimum HTTP inactivity timeout is 5 minutes.
- UPnP AV MediaServer control points invoke CMS:ConnectionComplete to terminate connections.

7.5.4.3.3.6.4

[GUIDELINE] If the http-stalling flag is true for a content binary, then the streaming HTTP Server Endpoint shall not change encoding rate based on the HTTP throughput.

[ATTRIBUTES]

M	A	DMS +PU+	M_DMS	n/a	[33]	OBPDK	N

NOTE: If a Rendering Endpoint has already opened a normal (1x) GET request, then it does not need to inform the server of the slow-motion playback since the Rendering Endpoint could simply consume the data more slowly. This is similar to the connectrion stalling approach to pause/resume, except that the client does not completely pause, it continues to consume and decode the content, but configures its decoder to do so at a slower rate. If the server supports stalling, then this is indistinguishable on the server side from a client that is doing pause/resume.

7.5.4.3.3.6.5

[GUIDELINE] If the http-stalling flag is true for a content binary, then the streaming HTTP Server Endpoint shall allow streaming at rates slower than real-time, while in streaming transfer mode, to support a Slow Forward Scan media operation using Connection Stalling to accommodate slower decode and display of the streamed content.

[ATTRIBUTES]

M	A	DMS +PU+	M_DMS	n/a	[33]	Q5ABD	N

7.5.4.3.3.7 MT HTTP Seek Media Operation

7.5.4.3.3.7.1

[GUIDELINE] An HTTP Client Endpoint should implement the Seek media operation by using the Range or TimeSeekRange.dlna.org header fields, if those header fields are supported by the HTTP Server for the content binary.

[ATTRIBUTES]

S	A	+DN+ +DNSYNC+	M-DMD	MIU	[33]	9I4UN	C

NOTE: This guideline recommends two ways to implement the seek operation on content. The method used by the client is conditional on whether or not the HTTP server supports the capability for that content.

See guideline 7.4.1.3.18 and 7.4.1.3.28 for information on discovering server seek capabilities.

See 7.5.4.3.2.22, and 7.5.4.3.2.24 guidelines for more information on Range and TimeSeekRange.dlna.org header fields.

7.5.4.3.3.7.2

[GUIDELINE] For every content binary not using DLNA Link Protection, if a streaming HTTP Client Endpoint performs a Seek media operation, then the HTTP Client Endpoint shall support all of the following Seek media operation methods:

- the Range header field
- the TimeSeekRange.dlna.org header field.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[33]	PHT47	N

NOTE: This guideline is applicable to audio Media Class and AV Media Class.

7.5.4.3.3.8 MT HTTP Fast Forward Scan Media Operation

7.5.4.3.3.8.1

[GUIDELINE] If an audio streaming HTTP Client Endpoint wants to perform a Fast Forward Scan media operation (positive play speed greater than 1x), then it should use one of these methods, provided that the given header fields are supported by the server.:

- Issuing multiple HTTP GET requests with a specified Range header field.
- Issuing multiple HTTP GET requests with a specified TimeSeekRange.dlna.org header field.
- Issuing a single HTTP GET request with a specified PlaySpeed.dlna.org header field.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[33]	TY7CW	C

NOTE: Forward Scan and backward Scan operations fall into a category of playback capabilities referred to as trick modes. These guidelines specify three general techniques for implementing trick modes.

The first technique is to issue multiple HTTP requests with specified byte ranges, such that the byte data can be rendered sequentially, giving the effect of a trick mode playback. With this technique, the HTTP Client endpoint is responsible for specifying the appropriate byte ranges that will achieve the desired effect.

The second technique is a variant of the first, and it works by requesting the HTTP server to return time ranges (instead of byte ranges). In this technique, the HTTP Client endpoint is responsible for choosing the appropriate time ranges that achieve the desired effect.

The third technique works by having the HTTP Client endpoint instruct the HTTP Server Endpoint to return byte data that is already time scaled for a particular play speed.

See 7.5.4.3.2.22 MT HTTP Header: Range (Server), 7.5.4.3.2.24 MT HTTP Time-Based Seek (Server), and 7.5.4.3.3.16 MT HTTP PlaySpeed.dlna.org Header guidelines for more information on Range, TimeSeekRange.dlna.org, and PlaySpeed.dlna.org header fields.

7.5.4.3.3.8.2

[GUIDELINE] For every AV content binary not using DLNA Link Protection, if a streaming HTTP Client Endpoint performs a Fast Forward Scan media operation (positive play speed greater than 1x), then the HTTP Client Endpoint shall support all of the following methods:

- Issuing multiple HTTP GET requests with a specified Range header field.
- Issuing multiple HTTP GET requests with a specified TimeSeekRange.dlna.org header field.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[33]	T24LR	N

7.5.4.3.3.8.3

[GUIDELINE] For every AV content binary not using DLNA Link Protection, if a streaming HTTP Client Endpoint performs a Fast Forward Scan media operation (positive play speed greater than 1x), then the HTTP Client Endpoint should support the following method:

- Issuing a single HTTP GET request with a specified PlaySpeed.dlna.org header field.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[33]	TYB9P	N

7.5.4.3.3.9 MT HTTP Streaming Slow Forward Scan Media Operation

7.5.4.3.3.9.1

[GUIDELINE] If an audio streaming HTTP Client Endpoint wants to perform a Slow Forward Scan media operation (positive play speed less than 1x), then it should use one of these methods:

- Issuing multiple HTTP GET requests with a specified Range header field
- Issuing a single HTTP GET request with a specified PlaySpeed.dlna.org header field.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[33]	L6RNZ	C

7.5.4.3.3.9.2

[GUIDELINE] For every AV content binary not using DLNA Link Protection, if a streaming HTTP Client Endpoint performs a Slow Forward Scan media operation (positive play speed less than 1x), then the HTTP Client Endpoint shall support all of the following methods:

- Issuing multiple HTTP GET requests with a specified Range header field
- Issuing a single HTTP GET request and subsequently using Connection Stalling Method (see 7.5.4.3.3.6) to accommodate slower decode and display of the streamed content.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[33]	Z9BD2	N

7.5.4.3.3.9.3

[GUIDELINE] For every AV content binary not using DLNA Link Protection, if a streaming HTTP Client Endpoint performs a Slow Forward Scan media operation (positive play speed less than 1x), then the HTTP Client Endpoint should support the following method:

- Issuing a single HTTP GET request with a specified PlaySpeed.dlna.org header field.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[33]	3W8KS	N

7.5.4.3.3.10 MT HTTP Streaming Fast Backward Scan Media Operation

7.5.4.3.3.10.1

[GUIDELINE] If an audio streaming HTTP Client Endpoint wants to perform a Fast Backward Scan operation (negative play speed less than -1x), then it should use one of these methods:

- Issuing multiple HTTP GET requests with a specified Range header field.
- Issuing multiple HTTP GET requests with a specified TimeSeekRange.dlna.org header field.
- Issuing a single HTTP GET request with a specified PlaySpeed.dlna.org header field.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[33]	Y7CWO	C

7.5.4.3.3.10.2

[GUIDELINE] For every AV content binary not using DLNA Link Protection, if a streaming HTTP Client Endpoint performs a Fast Backward Scan media operation (negative play speed less than -1x), then the HTTP Client Endpoint shall support all of the following methods:

- Issuing multiple HTTP GET requests with a specified Range header field.
- Issuing multiple HTTP GET requests with a specified TimeSeekRange.dlna.org header field.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[33]	V46LS	N

7.5.4.3.3.10.3

[GUIDELINE] For every AV content binary not using DLNA Link Protection, if a streaming HTTP Client Endpoint performs a Fast Backward Scan media operation (negative play speed less than -1x), then the HTTP Client Endpoint should support the following method:

- Issuing a single HTTP GET request with a specified PlaySpeed.dlna.org header field.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[33]	ZHSFA	N

7.5.4.3.3.11 MT HTTP Streaming Slow Backward Scan Media Operation

7.5.4.3.3.11.1

[GUIDELINE] If an audio streaming HTTP Client Endpoint wants to perform a Slow Backward Scan media operation (negative play speed greater than or equal to -1x), then it should use one of these methods:

- Issuing multiple HTTP GET requests with a specified Range header field., or
- Issuing a single HTTP GET request with a specified PlaySpeed.dlna.org header field.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[33]	6RNZ6	C

7.5.4.3.3.11.2

[GUIDELINE] For every AV content binary not using DLNA Link Protection, if a streaming HTTP Client Endpoint performs a Slow Backward Scan media operation (negative play speed less than zero and greater than or equal to -1x), then the HTTP Client Endpoint shall support the following methods:

- Issuing multiple HTTP GET requests with a specified Range header field

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[33]	A89O5	N

7.5.4.3.3.11.3

[GUIDELINE] For every AV content binary not using DLNA Link Protection, if a streaming HTTP Client Endpoint performs a Slow Backward Scan media operation (negative play speed less than zero and greater than or equal to -1x), then the HTTP Client Endpoint should support the following method:

- Issuing a single HTTP GET request with a specified PlaySpeed.dlna.org header field.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[33]	2DQOQ	N

7.5.4.3.3.12 MT HTTP Streaming Scan Media Operations

7.5.4.3.3.12.1

[GUIDELINE] If a streaming HTTP Client Endpoint wants to stop a normal playback stream in order to start a scan operation playback using the Range header under conditions where 7.5.4.3.2.4.6 applies, then it should close the original HTTP connection before issuing a GET request with the Range header to perform scan operations (a.k.a. trick modes). After closing the original connection, the streaming HTTP Client Endpoint should open a new HTTP connection for the scan operation. Please observe that the new HTTP connection might be a persistent connection, which allows the client to issue multiple GET requests on a single HTTP connection.

[ATTRIBUTES]

S	A	DMP DMR +DN+ +DNSYNC+	M-DMP	MIU	[33]	6TQT3	

NOTE: Note that transitions from scan operations to normal playback can be achieved by making open ended (i.e. last-byte-pos value in Range header is absent) GET requests with the Range option.

7.5.4.3.3.12.2

[GUIDELINE] If a streaming HTTP Client Endpoint wants to stop a normal playback stream in order to start a scan operation playback using the PlaySpeed.dlna.org header under conditions where 7.5.4.3.2.4.6 applies, then it should close the original HTTP connection before issuing a GET request with the PlaySpeed.dlna.org header to perform scan operations (a.k.a. trick modes). After closing the original connection, the streaming HTTP Client should open a new HTTP connection for the scan operation. Please observe that the new HTTP connection might be a persistent connection, which allows the client to issue multiple GET requests on a single HTTP connection.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[33]	M6TQT	

NOTE: Note that transitions from scan operations to normal playback can be achieved by closing the HTTP connection that provides scan operations and opening a new HTTP connection for normal playback speed.

7.5.4.3.3.13 MT HTTP Streaming Download Media Operation

[GUIDELINE] An HTTP Client Endpoint shall initiate a streaming download media operation with the HTTP GET method when using the HTTP transport protocol for content and it shall specify the transferMode.dlna.org HTTP header with a value of "Streaming".

[ATTRIBUTES]

M	L	+DN+ +DNSYNC+	M-DMD	MIU	[33]	X7UF4	

7.5.4.3.3.14 MT HTTP Prohibited Operations for Streaming Download

[GUIDELINE] An HTTP Client Endpoint performing a streaming download media operation shall not use any of the following media operations as specified in these guidelines.

Table 50 — HTTP Prohibited Operations References

Operation	Reference
Pause	7.5.4.3.3.4 MT HTTP Pause/Pause-Release Media Operation
Fast Forward Scan	7.5.4.3.3.8 MT HTTP Fast Forward Scan Media Operation
Slow Forward Scan	7.5.4.3.3.9 MT HTTP Streaming Slow Forward Scan Media Operation
Fast Backward Scan	7.5.4.3.3.10 MT HTTP Streaming Fast Backward Scan Media Operation
Slow Backward Scan	7.5.4.3.3.11 MT HTTP Streaming Slow Backward Scan Media Operation

[ATTRIBUTES]

M	L	+DN+ +DNSYNC+	M-DMD	MIU	[33]	AA772	

7.5.4.3.3.15 MT HTTP Media Operation Support Within a Profile

7.5.4.3.3.15.1

[GUIDELINE] If the following conditions are true, then the HTTP Client Endpoint shall support that media operation on all content with that media format profile.

- The HTTP Client Endpoint supports a media operation on a DLNA media format profile.
- The HTTP Server Endpoint supports the necessary protocol elements to support the media operation.

[ATTRIBUTES]

M	A	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	n/a	8H8X2	

NOTE: Necessary protocol elements can include optional parametes such as byte Range and TimeSeekRange.dlna.org.

7.5.4.3.3.15.2

[GUIDELINE] If an HTTP Server Endpoint supports byte Range and/or TimeSeekRange.dlna.org capabilities on the content for a particular DLNA media format profile, it should support that on all content with that media format profile.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	n/a	Z8H8X	

NOTE: In some cases, such as transcoded or live content, it can be difficult to support byte Range and/or TimeSeekRange.dlna.org.

7.5.4.3.3.16 MT HTTP PlaySpeed.dlna.org Header

7.5.4.3.3.16.1

[GUIDELINE] If the Content Source indicates support for the PlaySpeed.dlna.org HTTP header for a content binary (as defined in guidelines 7.4.1.3.21), then the HTTP Server Endpoint shall respond to PlaySpeed.dlna.org HTTP requests for that content binary with the clarifications and constraints defined in guidelines 7.5.4.3.3.16.2 - 7.5.4.3.3.16.6.

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	[33]	F57CP	

NOTE: The PlaySpeed.dlna.org HTTP header allows HTTP clients to request the HTTP server to return content in a time scaled form.

For example, a DLNA HTTP client can request a DLNA HTTP server to return DLNA content in a 4x playback speed. The HTTP server's response would send content that gives the appearance of 4x playback speed.

Also note that this HTTP header can be used by DLNA servers for content conforming to DLNA media format profiles and content that does not conform to DLNA media format profiles.

7.5.4.3.3.16.2

[GUIDELINE] If a streaming HTTP Server Endpoint receives the PlaySpeed.dlna.org HTTP header in a HEAD request, then the HTTP server may respond without the PlaySpeed.dlna.org HTTP header.

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	[33]	7UF4F

NOTE: HTTP servers are required to support the HEAD method. When an HTTP server gets a HEAD request with this option, the HTTP server can omit it from the response.

If HTTP clients need to determine if the PlaySpeed.dlna.org is supported for a URI, then the client are supposed to use the mandatory getcontentFeatures.dlna.org HTTP header and examine the DLNA.ORG_PS parameter.

The reason why HEAD responses do not require the use of PlaySpeed.dlna.org is that the computational effort to respond with only the headers for a request that includes PlaySpeed.dlna.org is significant.

7.5.4.3.3.16.3

[GUIDELINE] If the streaming HTTP Endpoint uses the PlaySpeed.dlna.org HTTP header, then the Endpoint shall use the following syntax for the HTTP header and its value.

- PlaySpeed-line = "PlaySpeed.dlna.org" *LWS ":" *LWS play speed specifier
- play speed specifier = "speed" "=" transport-play-speed
- transport-play-speed = <use the same notation for the AVT.TransportPlaySpeed state variable defined in the UPnP AV Transport service type>

Note that the "speed" token is case sensitive.

Examples:

- PlaySpeed.dlna.org : speed=10
- PlaySpeed.dlna.org : speed=-1/2

When encountering syntax errors with the PlaySpeed.dlna.org HTTP header, the HTTP server shall use the HTTP response code 400 (Bad Request).

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[33]	A772R

NOTE: The field value specifies a play speed to scale content data of a resource. The value is represented as same as TransportPlaySpeed state variable defined by AV Transport service type (e.g. 5, 10, -1/2. -10, -3/2, etc.).

7.5.4.3.3.16.4

[GUIDELINE] If the streaming HTTP Server Endpoint returns data (Target Response) for scaled content to be decoded for a scan operation (a.k.a. variable play speed) with the HTTP response code 200 (OK), then the HTTP response message shall specify the PlaySpeed.dlna.org HTTP header field to indicate the play speed of the scaled content.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	57CPW	

NOTE: This guideline requires the HTTP server to indicate if content bytes in the HTTP response represent content that has been time scaled.

7.5.4.3.3.16.5

[GUIDELINE] If the requested play speed is not valid for the resource with the URI specified in the HTTP GET request, then the HTTP server shall respond with the HTTP response error code 406 (Not Acceptable). Interpretation of not valid includes the following types of errors:

If an HTTP server supports the PlaySpeed.dlna.org header and the requested play speed is not valid for the resource with the URI specified in the HTTP GET request, then the HTTP server shall respond with the HTTP response error code 406 (Not Acceptable). Interpretation of not valid includes the following types of errors:

- The requested TransportPlaySpeed is not supported for the content.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	H8X2Y	

NOTE: This guideline specifies the error code to be used in scenarios where the HTTP server cannot accommodate a request to time scale content.

The HTTP Server Endpoint indicates support for the PlaySpeed.dlna.org header in the DLNA_ORG_PS parameter which is in the contentFeatures.dlna.org in guideline 7.4.1.3.21 MM ps-param (Server-Side PlaySpeeds Parameter).

7.5.4.3.3.16.6

[GUIDELINE] The scaled data (returned by the streaming HTTP Server Endpoint as a Target Response) shall be compliant to the media format profile indicated in the corresponding <res> element, obtained from a UPnP AV ContentDirectory service implementation.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33] [49]	7CPWA	

NOTE: This guideline obligates an HTTP Server Endpoint to return content bytes that are conformant to the characteristics described for that content by the associated UPnP AV ContentDirectory service. For example, if the content is exposed as MPEG2_PS_NTSC content, the returned content must conform to the MPEG syntax defined for that Media Format Profile.

7.5.4.3.3.16.7

[GUIDELINE] The scaled data stream (returned by the streaming HTTP Server Endpoint as a Target Response) may utilize different media format parameters than the original (unscaled) stream, as long as these differences do not impact its compliance with the media format profile of the original (unscaled) content binary.

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	[33] [49]	CSWJC	N

NOTE: This guideline provides considerable leeway in the Serving Endpoint's implementation of scaled content. For example, consider a Serving Endpoint A that implements a Scale value of 2 by dropping every other frame in a 30 fps stream resulting in a content stream that still contains 30 frames per second, but that appears to be playing at twice the normal speed. Next consider Serving Endpoint B that implements the same Scale value of 2 by sending 4 I-Frames a second with a presentation time of 250ms per frame, which will also appear to be playing twice as fast. Finally, consider a Serving Endpoint C that implements a Scale value of 2 by transcoding the original stream into a new stream with the same Format Profile, and which gives the appearance of playing twice as fast. In addition to dropping frames, servers may also choose to omit audio tracks, in order to minimize the bitrate of scaled streams.

7.5.4.3.3.16.8

[GUIDELINE] If the Content Source does not indicate support for the PlaySpeed.dlna.org HTTP header for a content binary (as defined in guideline 7.4.1.3.21), then the HTTP Client Endpoint shall not issue HTTP GET and HEAD requests with the PlaySpeed.dlna.org HTTP header.

[ATTRIBUTES]

M	L	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33]	ZAVEC	

7.5.4.3.3.16.9

[GUIDELINE] If the PlaySpeed.dlna.org request is syntactically correct and the requested play speed is valid, and if the HTTP Server Endpoint returns data including the requested play speed (Target Response) then it shall respond to the HTTP GET request with the 200 (OK) HTTP response code.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	9WS9T	

7.5.4.3.3.17 MT HTTP PlaySpeed Position

[GUIDELINE] If the Content Source indicates support for the PlaySpeed.dlna.org HTTP header for a content binary (as defined in guidelines 7.4.1.3.21), then the HTTP Server Endpoint may provide NPT and byte-pos (byte offset) information in chunk header when transferring data in chunked transfer when PlaySpeed is not equal to 1x. The NPT and byte-pos values in a chunk header ,should correspond to the chunk data. The format of chunk extension to provide NPT and byte-pos information shall be:

- Chunk extension to pass NPT information
  npt=npt-time as defined in guidelline 7.5.4.2.13 (npt sec | npt hhmmss)
- Chunk extension to pass byte offset:
  byte-pos=byte-offset-value

The format of chunk header with the above two chunk extensions is as follows:

- chunk-size“; npt=”npt-time“; byte-pos=”byte-offset-value CRLF

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	[33]	6U8UB	N

NOTE: The client endpoint uses this information to keep track of elapsed time accurately. This is specifically useful when there are discontinuities in PTS. One possible scenario where PTS discontinuities are expected is segmented recordings.

7.5.4.3.3.18 MT HTTP Header: chunckEncodingMod.dlna.org

7.5.4.3.3.18.1

[GUIDELINE] If a Serving Endpoint receives an HTTP Get request with ChunkEncodingMode.dlna.org header and PlaySpeed different than 1, the Serving Endpoint may send the response using HTTP chunked transfer coding.

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	[33]	J25AL	N

7.5.4.3.3.18.2

[GUIDELINE] The notation for the ChunkEncodingMode.dlna.org header is defined as follows:

- ChunkEncodingMode -line = "ChunkEncodingMode.dlna.org" * LWS":" 1

The only value possible is "1".

[ATTRIBUTES]

M	A	DMS +PU+ DMP DMR	M-DMS	n/a	[33]	ES67D	N

NOTE: Example is as follows:

- ChunkEncodingMode:1

7.5.4.3.3.18.3

[GUIDELINE] A receiving endpoint may use the ChunkEncodingMode.dlna.org header to request chunked transfer encoding from a serving endpoint.

[ATTRIBUTES]

O	A	DMP DMR	M-DMP	MIU	[33]	EUYJG	N

7.5.4.3.3.19 MT Combined Range, Time based Seek, and Play Speed HTTP Requests

7.5.4.3.3.19.1

[GUIDELINE] If a streaming HTTP Server Endpoint receives an HTTP GET request with both the PlaySpeed.dlna.org and the TimeSeekRange.dlna.org header fields, the endpoint shall understand that a scan operation is requested for the specified time range.

If the streaming HTTP Server Endpoint can never process either or both time-scaling and time-seek, the error code for time-scaling, 406(Not Acceptable) shall be returned.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	PWAEV	

NOTE: This guideline covers what a streaming HTTP server endpoint needs to do if it receives an HTTP request for both time based seek and play speed. This guideline also infers how a streaming HTTP Client endpoint can expect an HTTP Server Endpoint to behave.

7.5.4.3.3.19.2

[GUIDELINE] If a streaming HTTP Server Endpoint receives an HTTP GET request with Range and (TimeSeekRange.dlna.org or PlaySpeed.dlna.org) header fields, then the Range header field shall take the highest precedence and the server shall ignore the other time seek and play speed fields.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	CPWAE	

NOTE: This guideline covers what a streaming HTTP Server endpoint needs to do if it receives an HTTP request with Range and other DLNA fields for play speed or time based seek. This guideline also infers how a streaming HTTP Client endpoint can expect an HTTP Server Endpoint to behave.

7.5.4.3.3.20 MT HTTP Header: realTimeInfo.dlna.org

7.5.4.3.3.20.1

[GUIDELINE] HTTP Client Endpoints may use realTimeInfo.dlna.org in GET or HEAD requests to specify the desired policy for content delivery.

[ATTRIBUTES]

O	A	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	[33]	WAEVZ	

NOTE: By using this header, an HTTP Client Endpoint (e.g. DMP) informs an HTTP Server Endpoint (e.g. DMS) the way it wants to receive content byte data (unmodified or in a real-time manner). In the second case, the loss of content quality (i.e. drop data bytes) is possible.

In the case of real-time delivery, it provides a hint to an HTTP Client Endpoint on how long to pre-buffer content data bytes before beginning the playback.

7.5.4.3.3.20.2

[GUIDELINE] If an HTTP Client Endpoint request contains the realTimeInfo.dlna.org header, then the HTTP Server Endpoint reply should include the realTimeInfo.dlna.org header.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	[33]	2Y963	

NOTE: An HTTP Client Endpoint (e.g. DMP) must be ready to receive an undesired value for the max-delay-time from an HTTP Server Endpoint (e.g. DMS).

It is worth to mention that for a live content the max-lag-time value can't be set as infinite.

7.5.4.3.3.20.3

[GUIDELINE] If an HTTP Server Endpoint responds with a realTimeInfo.dlna.org header with an infinite max-lag-time (i.e. with value of "*"), then it shall not drop and/or modify any portion of content byte data.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	X2Y96	

7.5.4.3.3.20.4

[GUIDELINE] If an HTTP Server Endpoint responds with a realTimeInfo.dlna.org header with a finite max-lag-time, then it shall not send stale data when the delay time is more than the max-lag-time. In this case an HTTP Server Endpoint shall omit the Content-Length header from HTTP response.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	2R2GY	

7.5.4.3.3.20.5

[GUIDELINE] An HTTP Server Endpoint should include the realTimeInfo.dlna.org header in each HTTP reply.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	[33]	8X2Y9	

NOTE: It is desirable to provide an HTTP Client Endpoint (e.g. DMP) the information for the delivery manner each time.

7.5.4.3.3.20.6

[GUIDELINE] If an HTTP Server Endpoint receives an HTTP GET or HEAD request with a Range and realTimeInfo.dlna.org header, then an HTTP Server Endpoint shall never reply with a finite max-lag-time parameter value.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[33]	R2GYQ

NOTE: This guideline obliges an HTTP Server Endpooint to not change the content data bytes in a reply to a request with a Range header.

7.5.4.3.3.20.7

[GUIDELINE] The notation for the realTimeInfo.dlna.org HTTP header is defined as follows:

- realTimeInfo-line = "realTimeInfo.dlna.org" *LWS ":" *LWS max-lag-time
- max-lag-time= "DLNA.ORG_TLAG" "=" duration
- duration = npt-sec | "*"
- ntp-sec= 1*DIGIT["."1*3DIGIT]

Note that the literal, "DLNA.ORG_TLAG" is case sensitive.

The max-lag-time is the maximum allowed delay between the current time and the time at which a particular portion of content data shall be sent in order to meet the real-time delivery requirements for content. If the delay for a particular portion of content data exceeds the max-lag-time, then an HTTP Server Endpoint shall drop it instead of sending it. The value "*" indicates that the content data bytes will never expire. This guarantees that an HTTP Client Endpoint will receive all of the content data bytes.

Duration shall be given in seconds.

Example:

- realTimeInfo.dlna.org : DLNA.ORG_TLAG=1.75
- realTimeInfo.dlna.org : DLNA.ORG_TLAG=*

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+ +DN+ +PR1+ +PR2+ +DNSYNC+	M-DMS M-DMP M-DMD	MIU	[33]	72R2G

NOTE: The max-lag-time value provides information to an HTTP Client Endpoint (e.g. DMP) on how the content will be delivered. When it is infinite, all requested bytes of the original content will be delivered without any modifications (i.e. data loss). Otherwise, an HTTP Client Endpoint (e.g. DMP) can't expect to receive the content unmodified (i.e. can contain some data loss).

In the case for real-time delivery of content, the max-lag-time negotiation aids an HTTP Client Endpoint (e.g. DMP) to adjust its pre-buffering time and for an HTTP Server Endpoint (e.g. DMS) its delay buffer. This negotiation aids an HTTP Server Endpoint from sending stale content data bytes instead of up-to-date ones.

7.5.4.3.3.21 MT HTTP Media Operations Support

7.5.4.3.3.21.1

[GUIDELINE] If a media operation (play, stop, pause/release, seek, scan operations) is supported for a content binary, Content Receivers shall support the media operation for the entire known length of the content binary.

[ATTRIBUTES]

M	A	DMP DMR +DN+ +DNSYNC+	M-DMP M-DMD	MIU	n/a	772R2	

7.5.4.3.3.21.2

[GUIDELINE] If an op-param (7.4.1.3.18.1) capability (for play, stop, pause/release, seek, scan operations) is supported for a content binary, Content Sources shall support the media operation for the entire length of the content binary.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[56]	F658Q	

NOTE: In the case of time seek range, HTTP Server Endpoints are still allowed to align responses to frame boundaries as specified in DLNA__Media_Formats subclause 9.2.7.

7.5.4.3.4 HTTP Transport: Interactive Transfer Guidelines

7.5.4.3.4.1 MT HTTP Interactive Transfer Initiation

7.5.4.3.4.1.1

[GUIDELINE] An HTTP Client Endpoint requesting a content binary from an HTTP Server Endpoint with an Interactive Transfer shall use the HTTP GET method.

[ATTRIBUTES]

M	A	DMPr DMP DMR	M-DMP	MIU	[33]	QT357	

NOTE: Interactive Transfers with GET are used to get Image content for immediate rendering.

7.5.4.3.4.1.2

[GUIDELINE] If an HTTP Client Endpoint requests an Interactive Transfer, then it shall specify the transferMode.dlna.org HTTP header and the header shall have a value of "Interactive".

[ATTRIBUTES]

M	A	DMPr DMP DMR	M-DMP	MIU	[33]	NZ6EY	

NOTE: Version 1.0 clients will still request interactive transfers of images without using the transferMode.dlna.org HTTP header. Guideline 7.5.4.3.2.33.2 requires that servers that receive such a request treat it as equivalent to a request for an interactive transfer for image media classes.

7.5.4.3.4.1.3

[GUIDELINE] If an HTTP Server Endpoint receives an HTTP GET or HEAD request with a transferMode.dlna.org HTTP header with the value of "Interactive" and the requested URI supports Interactive Transfer Mode (as defined in 7.4.1.3.35.1), then the Target Response shall replicate this HTTP header.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	UF4F6

7.5.4.3.4.1.4

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						TQT35

7.5.4.3.4.2 MT Interactive Transfer headers interactions

7.5.4.3.4.2.1

[GUIDELINE] An HTTP Client Endpoint shall not use the following headers when requesting an Interactive Transfer:

- TimeSeekRange.dlna.org
- PlaySpeed.dlna.org
- realTimeInfo.dlna.org

[ATTRIBUTES]

M	A	DMPr DMP DMR	M-DMP	MIU	[33]	4F658

NOTE: See 7.5.4.3.2.24 MT HTTP Time-Based Seek (Server), and 7.5.4.3.3.16 MT HTTP PlaySpeed.dlna.org Header guidelines for more information on the operations of these headers

7.5.4.3.4.2.2

[GUIDELINE] This guideline no longer applies.

[ATTRIBUTES]

						T3577

7.5.4.3.4.3 MT Range Behavior for Interactive Transferred Content

7.5.4.3.4.3.1

[GUIDELINE] The Range HTTP header may be used for Interactive Transfers.

[ATTRIBUTES]

O	C	DMS DMPr DMR DMP +PU+ +PR1+ +PR2+	M-DMS M-DMP	MIU	[33]	F4F65

NOTE: The behavior and usage model for the Range header are governed by other guidelines. Specifically:

- 7.4.1.3.18 MM op-param (Operations Parameter - Common Guidelines) describes how the op-param indicates whether the Range header is supported under the "Full Random Access Data Availability" model.
- 7.4.1.3.28 MM lop-npt, lop-bytes and lop-cleartextbytes (Limited Operations Flags): Common describes how lop-bytes indicates whether the Range header is supported under the "Limited Random Access Data Availability" model.
- 7.5.4.3.2.18 MT HTTP Common Random Access Data Availability Requirements, 7.5.4.3.2.19 MT HTTP Data Range of "Full Random Access Data Availability", and 7.5.4.3.2.20 MT HTTP: Data Range of "Limited Random Access Data Availability" (MT) explain details about both models.

Interactive Transfers involving stored content generally fall under the category of "Full Random Access Data Availability" model. In some cases, an Interactive Transfer involving converted content will fall under the "Limited Random Access Data Availability" model when the mode-flag is 1.The "Limited Random Access Data Availability" model when the mode-flag is 0 does not apply to content that is transferred with Interactive Mode because that content is not time based.

7.5.4.3.5 HTTP Transport: Background Transfer Guidelines

7.5.4.3.5.1 MT HTTP Background Transfer Initiation

7.5.4.3.5.1.1

[GUIDELINE] An HTTP Client Endpoint downloading a content binary from an HTTP Server Endpoint with a Background Transfer request shall use the HTTP GET method.

[ATTRIBUTES]

M	A	DMPr +DN+ +DNSYNC+	M-DMD	MIU	[33]	3577T	

7.5.4.3.5.1.2

[GUIDELINE] An HTTP Client Endpoint uploading a content binary to an HTTP Server Endpoint with a Background Transfer request shall use the HTTP POST method.

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[33]	Z6EYR	

7.5.4.3.5.1.3

[GUIDELINE] If an HTTP Client Endpoint requests a Background Transfer, then it shall specify the transferMode.dlna.org HTTP header and the header shall have a value of "Background".

[ATTRIBUTES]

M	A	DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMD M-DMU	MIU	[33]	RNZ6E	

7.5.4.3.5.1.4

[GUIDELINE] If an HTTP Server Endpoint receives an HTTP GET or HEAD request with a transferMode.dlna.org HTTP header with the value of "Background" and the requested URI supports the Background Transfer Mode (as defined in 7.4.1.3.36), then the Target Response shall replicate this HTTP header.

[ATTRIBUTES]

M	A	DMS +PU+ +PR1+ +PR2+	M-DMS	n/a	[33]	6EYR3	

7.5.4.3.5.1.5

[GUIDELINE] Guideline no longer applies.

[ATTRIBUTES]

						OVY6R	

7.5.4.3.5.2 MT Background Transfer header interactions

7.5.4.3.5.2.1

[GUIDELINE] An HTTP Client Endpoint shall not use the following headers when requesting a Background Transfer with either the GET or POST methods.

- TimeSeekRange.dlna.org
- PlaySpeed.dlna.org
- realTimeInfo.dlna.org

[ATTRIBUTES]

M	A	DMPr +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMD M-DMU	MIU	[33]	7CWOV

NOTE: See 7.5.4.3.2.24 MT HTTP Time-Based Seek (Server), and 7.5.4.3.3.16 MT HTTP PlaySpeed.dlna.org Header guidelines for more information on the operations of these headers

7.5.4.3.5.2.2

[GUIDELINE] This guideline no longer applies.

[ATTRIBUTES]

						WOVY6

7.5.4.3.5.3 MT Range Behavior for Background Transferred Content

[GUIDELINE] The Range HTTP header may be used for Background Transfers.

[ATTRIBUTES]

O	C	DMS DMPr +PU+ +PR1+ +PR2+ +DN+ +UP+ +DNSYNC+ +UPSYNC+	M-DMS M-DMD M-DMU	MIU	[33]	CWOVY

NOTE: The behavior and usage model for the Range header are governed by other guidelines. Specifically:

- 7.4.1.3.18 MM op-param (Operations Parameter - Common Guidelines) describes how the op-param indicates whether the Range header is supported under the "Full Random Access Data Availability" model.
- 7.4.1.3.28 MM lop-npt, lop-bytes and lop-cleartextbytes (Limited Operations Flags): Common describes how lop-bytes indicates whether the Range header is supported under the "Limited Random Access Data Availability" model.
- 7.5.4.3.2.18 MT HTTP Common Random Access Data Availability Requirements, 7.5.4.3.2.19 MT HTTP Data Range of "Full Random Access Data Availability", and 7.5.4.3.2.20 MT HTTP: Data Range of "Limited Random Access Data Availability" (MT) explain details about both models.

Background Transfers involving stored content generally fall under the category of "Full Random Access Data Availability" model. In some cases, a Background Transfer involving converted content will fall under the "Limited Random Access Data Availability" model when the mode-flag is 1. The "Limited Random Access Data Availability" model when the mode-flag is 0 is not recommended for content that is transferred with the Background Mode since the server response cannot be guaranteed to be accurate for "live contents" when using the Range HTTP header, as described in 7.5.4.2.16.2.

7.5.4.3.6 HTTP Transport: POST Guidelines

7.5.4.3.6.1 MT Content Management HTTP POST Content Acqusition

7.5.4.3.6.1.1

[GUIDELINE] An Upload Controller, Upload Synchronization Controller, M-DMU, or the MIU shall implement a content transfer process by using an HTTP/1.1 POST request.

[ATTRIBUTES]

M	L	+UP+ +UPSYNC+	M-DMU	MIU	[33]	I4UNM	

NOTE: See 7.4.1.7.26 for more information.

7.5.4.3.6.1.2

[GUIDELINE] HTTP Client Endpoints shall not use HTTP POST with an HTTP/1.0 indicator.

[ATTRIBUTES]

M	L	+UP+ +UPSYNC+	M-DMU	MIU	[33]	57N7M	

NOTE: HTTP POST is defined only for HTTP/1.1.

7.5.4.3.6.1.3

[GUIDELINE] An Upload Controller, Upload Synchronization Controller, M-DMU, or the MIU requesting a content transfer process by using the HTTP/1.1 POST request, should set the transferMode.dlna.org HTTP header to a value of "Background" in the request.

[ATTRIBUTES]

S	L	+UP+ +UPSYNC+	M-DMU	MIU	[33]	UNMYS	

NOTE: Values other than "Background" can be used when uploading certain contents, such as "live" content.

7.5.4.3.6.1.4

[GUIDELINE] A DMS or M-DMS that implements the Upload Device Option (i.e. the upload AnyContainer) as indicated by the <dlna:X_DLNACAP> element described in guideline 7.4.1.7.4.1), then it shall accept HTTP POST requests for receiving content in content transfer operation.

[ATTRIBUTES]

M	R	DMS	M-DMS	n/a	[33]	NMYS5	

NOTE: This is required by the ContentDirectory specification as the baseline manner of transferring content to a UPnP AV MediaServer.

7.5.4.3.6.1.5

[GUIDELINE] The HTTP Client Endpoint shall issue its first HTTP POST request (to upload content) within 30 seconds of the CDS:CreateObject response.

[ATTRIBUTES]

M	L	+UP+ +UPSYNC+	M-DMU	MIU	[33]	POVK7	

NOTE: See 7.4.1.7.26 for more information.

7.5.4.3.6.1.6

[GUIDELINE] If an HTTP Client Endpoint is using a persistent connection to perform multiple content transfer processes, then the HTTP client should issue subsequent HTTP POST requests within 5 minutes of the previous HTTP POST request's completion.

[ATTRIBUTES]

S	C	+UP+ +UPSYNC+	M-DMU	MIU	[33]	4UNMY	

NOTE: M-DMUs, Upload Controllers, or Upload Synchronization Controllers are urged to use persistent HTTP connections to upload multiple content binaries. Ideally, these content transfers happen with little delay between them. However, HTTP clients are discouraged from exceeding 5 minutes of inactivity between transfers.

7.5.4.3.6.1.7

[Guideline] The HTTP Server Endpoint that is receiving an HTTP POST request is free to terminate the connection at any time by closing the TCP connection.

[Attributes]

O	C	DMS	M-DMS	n/a	[33]	7N7MB

NOTE: DMS or M-DMS devices implement this behavior to indicate an error during the content transfer. For example, if the DMS no longer has enough space, it can terminate the TCP connection.

A DMS that does a TCP disconnection can automatically destroy CDS objects or <res> elements. However if possible and appropriate, the DMS is encouraged to allow an Upload Controller to retry a content transfer process, by not destroying the CDS object or <res> element.

7.5.4.3.6.1.8

[Guideline] The HTTP Server Endpoint that supports HTTP POST requests shall support requests that are encoded with *Chunked Transfer Coding*.

[Attributes]

M	R	DMS	M-DMS	n/a	[33]	G857N

NOTE: HTTP clients can use either the default HTTP message encoding or *Chunked Transfer Coding*.

7.5.4.3.6.1.9

[Guideline] The HTTP Client Endpoint shall provide the EXPECT HTTP header field with the value of "100-continue" in the request.

[Attributes]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[33]	857N7

NOTE: The subclause 8.2.3 Use of the 100 (Continue) Status in [33] defines the behavior when a HTTP client uses "Expect: 100-continue" in the request.

The HTTP client waits to send POST message body, i.e. a content binary, until it receives a response to the request.

7.5.4.3.6.1.10

[Guideline] If an HTTP Server Endpoint receives a POST request without the "EXPECT: 100-continue", then it should return an HTTP error of 400 (Bad Request) and terminate the TCP connection.

[Attributes]

S	A	DMS	M-DMS	n/a	[33]	QPOVK

7.5.4.3.6.1.11

[Guideline] If an HTTP Server Endpoint that receives a POST request's HTTP headers and the HTTP server cannot accept the POST request's message body, then the HTTP Server Endpoint shall not return an HTTP status code of 100 (Continue).

[Attributes]

M	R	DMS	M-DMS	n/a	[33]	5QPOV

NOTE: The guideline 7.5.4.3.6.1.13.and 7.5.4.3.6.1.14 provides examples of error cases.

7.5.4.3.6.1.12

[GUIDELINE] An HTTP Client Endpoint that uses *Chunked Transfer Coding* shall use a zero-length chunk to indicate the end of the content binary.

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[33]	T6KSW	

NOTE: DMS and M-DMS devices look for the zero-length chunk to know if the content binary was completely sent.

7.5.4.3.6.1.13

[GUIDELINE] If the HTTP Server Endpoint cannot accept an HTTP POST request due to the processing capacity or current state of the device, then the HTTP Server Endpoint should respond with an HTTP status code of 503 (Service Unavailable).

[ATTRIBUTES]

S	R	DMS	M-DMS	n/a	[33]	OVK7Y	

7.5.4.3.6.1.14

[GUIDELINE] If the HTTP Server Endpoint cannot accept an HTTP POST request due to the lack of storage capacity, then the HTTP Server Endpoint should respond with an HTTP status code 507 (Insufficient Storage).

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[33]	TQT6K	

NOTE: This error code aligns with RFC2518. DLNA does not use WebDAV as a normative reference.

7.5.4.3.6.1.15

[GUIDELINE] If the HTTP Server Endpoint receives an entire entity body, then the HTTP Server Endpoint shall respond with either an HTTP status code of 200 (OK) or an HTTP status code of 201 (Created).

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[33]	QT6KS	

7.5.4.3.6.1.16

[GUIDELINE] If an HTTP Server Endpoint rejects a POST entity body in the middle of data transfer (due to the processing capacity, current status of the device, storage capacity etc,), then the HTTP server may terminate the TCP connection.

[ATTRIBUTES]

O	R	DMS	M-DMS	n/a	[33]	8VREB	

NOTE: Vendors can determine whether or not data is actually stored.

7.5.4.3.6.1.17

[GUIDELINE] If an HTTP Server Endpoint rejects a POST entity body in the middle of data transfer, then it should also send an HTTP error response right before terminating the TCP connection.

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[33]	EBUTB	

NOTE: HTTP error responses allow the HTTP server to provide more details on the cause of the error. It is imperative that the HTTP server close the TCP connection after sending the error response because the error can interfere with the logic associated with pipelined HTTP requests.

The requirement for HTTP clients to be tolerant of these scenarios means that the HTTP client can gracefully handle the TCP disconnect. (i.e. devices that crash as a result of a TCP disconnect exhibit bad behavior) There is no requirement that the HTTP client make the HTTP error response available to the user.

HTTP clients can use pipelined POST requests, but these guidelines do not recommend their usage because the Upload System Usage model favors a model where an Upload Controller uses a set of CDS:CreateObject and HTTP POST transactions, rather than a set of CDS:CreateObject requests followed by a set of HTTP POST transactions.

Lastly, clients are encouraged to terminate the connection after receiving the POST response from the server because HTTP/1.1 POST transactions operate under persistent connection rules.

7.5.4.3.6.1.18

[GUIDELINE] If an HTTP Server Endpoint rejects a POST entity body in the middle of data transfer by sending an HTTP error response, then it shall terminate the TCP connection after sending the error response.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[33]	VREBU	

7.5.4.3.6.1.19

[GUIDELINE] HTTP Client Endpoints shall be tolerant of a TCP layer disconnect during an HTTP POST transaction, including those scenarios where the HTTP Server Endpoint sends an HTTP error response before the HTTP Client Endpoint has finished transmitting the POST request's message-body.

[ATTRIBUTES]

M	L	+UP+ +UPSYNC+	M-DMU	MIU	[33]	BUTB4	

7.5.4.3.6.1.20

[GUIDELINE] If the HTTP Server Endpoint detects or initiates a TCP disconnect during a content transfer process that does not support the resume content transfer option, then the MediaServer shall be capable of accepting the same upload AnyContainer or OCM: upload content request that created the CDS object, which has the failed content transfer.

[ATTRIBUTES]

M	A	DMS	M-DMS	MIU	n/a	KSWAC	

NOTE: In other words, the DMS is expected to not return an UPnP error in response to a new CDS:CreateObject request when the Upload Controller attempts to retry a failed upload AnyContainer or OCM: upload content operation.

A DMS can implement a variety of behaviors to comply with this guideline.

- The DMS can immediately destroy the CDS object associated with the failed content transfer.
- The DMS can leave the CDS object in the CDS hierarchy for 30 minutes or less, from the point of the failure. The CDS:CreateObject response returns a new CDS object.
- The DMS can leave the CDS object in the CDS hierarchy for 30 minutes or less, from the point of the failure. The CDS:CreateObject response returns the same CDS object because the metadata specified in the request is exactly the same.

7.5.4.3.6.1.21

[GUIDELINE] When the HTTP Client Endpoint detects a TCP disconnection before receiving the final response to the HTTP POST request, it shall not assume that the HTTP Server Endpoint will store the transferred portion of the entity body. Specifically, this means the following.

- If the HTTP Client Endpoint wants to retry an upload process without using the resume content transfer or retry IFO attempt features, it shall start completely over by doing the CDS:CreateObject request (as part of the upload AnyContainer or OCM: upload content operation) and following up with a new content transfer process. (Note that retry IFO attempt applies only to the transfer of IFO files and only is available when resume content transfer is also available.)
- If the HTTP Client Endpoint wants to use the resume content transfer operation, then it shall specify a first-byte-pos of res@dlna:uploadedSize for the Content-Range header in an HTTP POST request. In this case, CDS:CreateObject is not invoked.

[ATTRIBUTES]

M	C	+UP+ +UPSYNC+	M-DMU	MIU	[33]	VK7YF

NOTE: If resume content transfer is not supported and the HTTP Client attempts to retry the HTTP POST transaction without invoking CDS:CreateObject again, then the retry attempt can fail. This type of a retry attempt is not covered by the DLNA guidelines.

7.5.4.3.6.2 MT HTTP Content Transfer Error Detection During HTTP POST

7.5.4.3.6.2.1

[GUIDELINE] If an HTTP Server Endpoint observes a TCP disconnect under all of the following conditions, then the HTTP server shall assume that the HTTP Client Endpoint failed to transfer the content.

- HTTP POST request does not use *Chunked Transfer Coding*
- The number of received bytes is less than the specified Content-Length.

If an HTTP Server Endpoint detects that the HTTP Client Endpoint failed to transfer the content and more than 35 minutes has elapsed since the end of the failed transfer attempt, then the MediaServer is obligated by 7.4.1.7.28 MM/CM: Auto-Destroy Behavior for a Failed or Partial Content Transfer Process to automatically destroy the created CDS object.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[33]	N7MBJ

NOTE: HTTP Client Endpoints are required to use the Content-Length in HTTP POST requests that do not use *Chunked Transfer Coding*.

Guideline 7.5.4.3.2.15 clarifies the uses of Content-Length for media transfers.

7.5.4.3.6.2.2

[GUIDELINE] An HTTP Server Endpoint receiving a POST request shall observe at least 30 seconds of data inactivity before closing the TCP connection and assuming that the HTTP Client Endpoint failed to transfer the content.

Data inactivity is defined as the HTTP Server Endpoint not receiving content data from the HTTP Client Endpoint even though there is an established TCP connection.

This guideline works in conjunction with 7.5.4.3.6.1.7 because this guideline assumes Ideal Network Conditions and also assumes that neither user-initiated causes nor out-of-band system events cause the connection to be closed.

[Attributes]

M	A	DMS	M-DMS	n/a	[33]	MYS54	

NOTE: These guidelines specify a 5 minute timeout for stalled content transfer.

Note that the 7.5.4.3.6.1.7 permits HTTP Servers to close the connection at any time.

7.5.4.3.6.3 MT Client Content-Range

7.5.4.3.6.3.1

[Guideline] An HTTP Client Endpoint that uses the resume content transfer operation shall specify the Content-Range HTTP header field in a POST request to specify the range of content data sent in the request.

[Attributes]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[33]	VY6R4	

NOTE: The Content-Range header is defined in RFC-2616, subclause 14.16.

7.5.4.3.6.3.2

[Guideline] An HTTP Client Endpoint that uses the resume content transfer operation shall include and specify the instance-length parameter on the Content-Range HTTP header and last-byte position on the Content-Range HTTP header shall be instance-length-1.

[Attributes]

M	L	+UP+ +UPSYNC+	M-DMU	MIU	[33]	EYR3R	

NOTE: The instance-length parameter (see RFC-2616, subclause 14.16) specifies the total size of the object being uploaded.

The instance-length parameter must specify valid value, so "*" must not used for the instance-length parameter.

This guideline applies even when the HTTP client uses chunked transfer coding.

7.5.4.3.6.3.3

[Guideline] If an HTTP Client Endpoint specifies the Content-Range HTTP header the first-byte-pos shall be equal to the content length already stored in a peer HTTP server.

[Attributes]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[33]	577T6	

NOTE: Length of already stored data (on a peer HTTP server) can be known through the res@dlna:uploadedSize in a CDS:Browse response.

The first-byte-pos parameter is defined in RFC-2616, subclause 14.16.

7.5.4.3.6.3.4

[Guideline] An HTTP client shall use the Content-Range HTTP header only with POST requests that are part of a resume content transfer operation (i.e. use with GET nor HEAD request is prohibited).

[Attributes]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[33]	658QX	

7.5.4.3.6.3.5

[GUIDELINE] An HTTP Client Endpoint shall not specify the Content-Range HTTP header for a POST request addressed to res@importUri included in an object that does not support the resume content transfer functionality.

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[33]	2GYQ2

NOTE: An Upload controller, Upload Synchronization Controller, MIU, or M-DMU is able to know if the MediaServer supports resume functionality by res@dlna:resumeUpload in CDS:CreateObject.

7.5.4.3.6.3.6

[GUIDELINE] An HTTP Client Endpoint shall not specify the Content-Range HTTP header for a POST request addressed to res@dlna:importIfoFileURI.

[ATTRIBUTES]

M	A	+UP+ +UPSYNC+	M-DMU	MIU	[33]	Y9634

NOTE: The resume content transfer operation can be supported for res@importUri and cannot be used for res@dlna:importIfoFileURI. To recover from a failed transfer of an IFO file, the HTTP client simply does a retry IFO attempt, which is an HTTP POST request without the Content-Range HTTP header. Please note that type of transfer recovery is out of scope for res@importUri values.

7.5.4.3.6.4 MT Server Receiving Content-Range

7.5.4.3.6.4.1

[GUIDELINE] This guideline no longer applies.

[ATTRIBUTES]

						AEVZ5

7.5.4.3.6.4.2

[GUIDELINE] This guideline no longer applies.

[ATTRIBUTES]

						EVZ5S

7.5.4.3.6.4.3

[GUIDELINE] If an HTTP Server Endpoint has resume functionality and receives a POST request with Content-Range addressed to res@importUri which includes a syntax error, it should respond an HTTP response which status code is 400 (Bad Request).

[ATTRIBUTES]

S	A	DMS	M-DMS	n/a	[33]	VZ5SV

NOTE: If an instance-length is "*", in the Content-Range, DMS is expected to respond with an HTTP response with a status code of 400.

7.5.4.3.6.4.4

[GUIDELINE] If an HTTP Server Endpoint has resume functionality and receives a POST request with the Content-Range header in which the content-range-spec has a first-byte-pos value that is equal to the object byte size already stored (i.e. res@dlna:uploadedSize), then the HTTP Server Endpoint shall append the incoming uploaded data to the stored one.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[33]	96343	

NOTE: The Content-Range header and the first-byte-pos parameter are defined in RFC-2616, subclause 14.16.

A client can get the value of res@dlna:uploadedSize from a browse operation after the failure.

7.5.4.3.6.4.5

[GUIDELINE] If an HTTP Server Endpoint has resume functionality and receives a POST request with the Content-Range header in which the content-range-spec has a first-byte-pos value that is not equal to the object byte size already stored, then the HTTP server shall send an HTTP response which status code is 409 (Conflict).

This guideline is applied only when other error condition is not satisfied.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[33]	6343S	

7.5.4.3.6.5 MT HTTP POST Pipelining

7.5.4.3.6.5.1

[GUIDELINE] If an HTTP Client Endpoint initiates pipelined HTTP POST transactions, a subsequent pipelined POST request shall happen after the previous message-body has been sent completely.

[ATTRIBUTES]

M	C	+UP+ +UPSYNC+	M-DMU	MIU	[33]	YQ2KA	

NOTE: An Upload Controller or Upload Synchronization Controller that starts a subsequent POST transaction before having sent the message-body for the previous POST transaction will confuse the HTTP Server Endpoint into thinking the subsequent POST transaction is the message-body for the first POST transaction.

7.5.4.3.6.5.2

[GUIDELINE] An HTTP Server Endpoint may terminate the HTTP session after responding to a POST request with the 200 (OK) or 201 (Created) responses.

[ATTRIBUTES]

O	C	DMS	M-DMS	n/a	[33]	GYQ2K	

NOTE: DMS devices are not required to support pipelined HTTP POST transactions. Therefore, it remains the responsibility of the Upload Controller to retry an attempted pipelined POST transacations when the HTTP server does not support pipelined POST transactions.

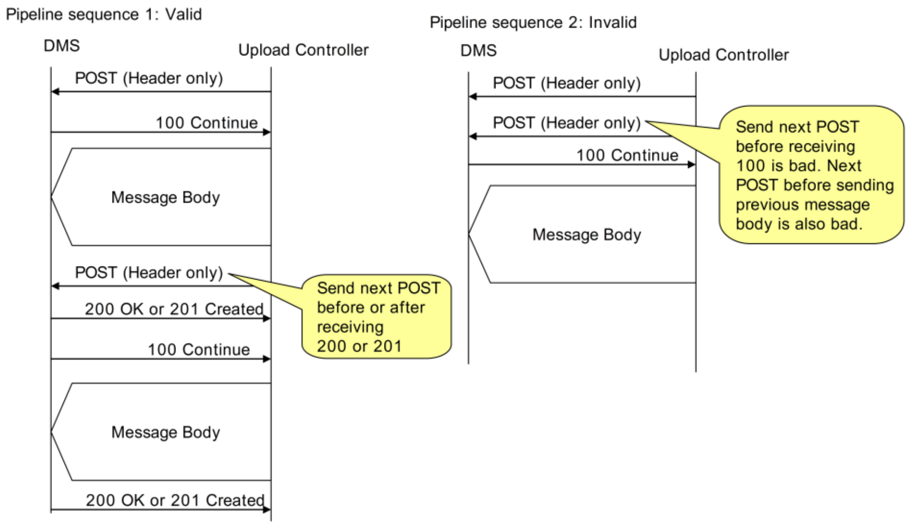


Figure 33 — Example of a valid and invalid pipelined POST transaction

7.5.4.4 RTP Media Transport

7.5.4.4.1 General

The Real-time Transport Protocol (RTP) [38] together with its companion protocol (RTCP) constitutes the basis of a transport mechanism for real-time media streams. In DLNA, RTP is used in combination with the RTSP protocol [27], SDP protocol [28], RTP payload formats and their associated media format profiles. These protocols define the DLNA RTP media transport which is optionally supported in DLNA, in addition to the required HTTP transport.

The subclauses in this section define guidelines for the implementation of the RTP media transport in the context of home networking. These apply to the playback of DLNA content and the streaming transactions between DLNA device classes. These guidelines do not specify behavior for non-DLNA device entities. A device class can be implemented by software running on a more general-purpose device/platform. For example, the RTP server of a Serving Endpoint can be used to serve DLNA and non-DLNA content. These guidelines would apply only when the DLNA-content is being served.

For these DLNA guidelines, RTP/RTCP is a protocol which is carried over UDP/IPv4 for unicast connections. A graphical representation of the reference protocol stack for the RTP media transport is shown in Annex H. The RTP media transport is enriched with the support of advanced features. For example, RTP retransmission can be used to increase reliability of media delivery. The RTP media transport also supports mechanisms for adaptive media delivery to improve continuous playback even in adverse network conditions. These guidelines also cover the RTSP protocol, which is carried over TCP connections. RTSP supports pause, seek and scan media operations (fast forward, slow forward, fast backward and slow backward).

Lastly, the DLNA guidelines do not specify interoperability for the transfer of content for either the Upload System Usage or the Download System Usage. Likewise, the DLNA guidelines also assume that the RTP Media Transport applies only to the Streaming Transfer Mode. The Interactive and Background Transfer modes do not apply to the RTP Media Transport.

The guidelines for RTP are organized as follows:

- RTP/RTCP Protocols: 7.5.4.4.2 RTP Media Transport: RTP/RTCP Protocols provides general RTP media transport guidelines. 7.5.4.4.3 RTP Serving Endpoint Requirements provides RTP Serving Endpoints guidelines, while 7.5.4.4.4 RTP Receiving Endpoint Requirements provides RTP guidelines for Receiving Endpoints. These subclauses also include guidelines for RTCP.
- Adaptation of media format profiles 7.5.4.4.5.2 A/V Media Format Profiles in the context of RTP transmission through 7.5.4.4.5.11 Guidelines for the encapsulation AMR-WBplus streams provides guidelines on adaptating several media format profiles for use with RTP and RTP payload formats.
- RTSP/SDP Protocols: 7.5.4.4.6.2 RTP Media Transport: RTSP/SDP Protocols includes guidelines for the support of the Real-Time Streaming Protocol [27] and the support for the Session Description Protocol [28].

Note that the terms Serving Endpoint and Receiving Endpoint are specific to RTP[5] , and they are defined specifically in 7.5.4.4.2.3 MT RTP on Serving Endpoints and 7.5.4.4.2.4 MT RTP on Receiving Endpoints. Respectively, they represent the different RTP, RTCP, RTSP, and SDP server/client components needed for a Content Source or Content Receiver using RTP.

7.5.4.4.2 RTP Media Transport: RTP/RTCP Protocols

7.5.4.4.2.1 MT RTP Optional Transport: RTP

[Guideline] DLNA devices may implement RTP over UDP as an optional media transport. guidelines in

- 7.5.4.4.2 RTP Media Transport: RTP/RTCP Protocols
- 7.5.4.4.3 RTP Serving Endpoint Requirements
- 7.5.4.4.4 RTP Receiving Endpoint Requirements
- 7.5.4.4.5.2 A/V Media Format Profiles in the context of RTP transmission
- 7.5.4.4.5.3 Adaptation of Audio-only Media Format Profiles to RTP, and Adaptation of the Audio component of A/V Media Format Profiles when transmitted as separate elementary streams
- 7.5.4.4.5.4 Guidelines for encapsulation of MPEG-2 streams
- 7.5.4.4.5.5 Guidelines for encapsulation of WMA and WMV elementary streams
- 7.5.4.4.5.6 Guidelines for the encapsulation for AVC elementary streams
- 7.5.4.4.5.7 Guidelines for the encapsulation for MPEG-4 part 2 elementary streams
- 7.5.4.4.5.8 Guidelines for the encapsulation of MPEG-4 AAC streams
- 7.5.4.4.5.9 Guidelines for the encapsulation of H.263 streams
- 7.5.4.4.5.10 Guidelines for the encapsulation AMR streams
- 7.5.4.4.5.11 Guidelines for the encapsulation AMR-WBplus streams, and
- 7.5.4.4.6.2 RTP Media Transport: RTSP/SDP Protocols

only when the RTP media transport is implemented.

[Attributes]

O	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	n/a	8QX8V

NOTE: RTP is an optional media transport. If RTP is optionally implemented by Serving Endpoints and Receiving Endpoints, the Guidelines that follow in these Tables must be implemented as described.

5. In v1.0 of the DLNA guidelines, serving endpoint and rendering endpoint were used. Since the system usages for v1.5 is greatly expanded, those terms were deemed insufficient to represent the different content roles on the network.

RTP is recommended for certain applications e.g. Streaming Transfer of live content.

7.5.4.4.2.2 MT RTP Applicable media class

[GUIDELINE] Serving Endpoints and Receiving Endpoints using this media transport shall use it only for Audio or Audio/Video media classes

[ATTRIBUTES]

M	L	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	n/a	58QX8	

7.5.4.4.2.3 MT RTP on Serving Endpoints

[GUIDELINE] Serving Endpoints shall comply with required features of RFC 3550 with constraints and additions as specified in 7.5.4.4.3 RTP Serving Endpoint Requirements.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[38]	7T63U	

NOTE: Serving Endpoints in DLNA have no need to receive RTP packets.

7.5.4.4.2.4 MT RTP on Receiving Endpoints

7.5.4.4.2.4.1

[GUIDELINE] Receiving Endpoints shall comply with required features of [38] (RFC 3550) with constraints and additions as specified in 7.5.4.4.4 RTP Receiving Endpoint Requirements.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[38]	R3RVS	

NOTE: Receiving Endpoints in DLNA have no need to transmit RTP packets.

7.5.4.4.2.5 MT RTP RTSP/SDP

7.5.4.4.2.5.1

[GUIDELINE] Serving Endpoints and Receiving Endpoints shall use RTSP and SDP as defined in 7.5.4.4.6.2 RTP Media Transport: RTSP/SDP Protocols.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	n/a	77T63	

NOTE: Certain RTP session parameters, such as UDP port, RTP clock frequency and RTP payload type, must be communicated before RTP packets can be exchanged. These parameters must be communicated using RTSP and SDP as defined in 7.5.4.4.6.2 RTP Media Transport: RTSP/SDP Protocols.

7.5.4.4.2.6 MT RTP Profile Support

7.5.4.4.2.6.1

[GUIDELINE] The following RTP profile shall be supported by Serving Endpoints and Receiving Endpoints:

RTP Profile for Audio and Video Conferences with Minimal Control as defined in RFC 3551, also called RTP/AVP.

[ATTRIBUTES]

M	L	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[38] [39]	YR3RV

7.5.4.4.2.6.2

[GUIDELINE] The following RTP profile may optionally be supported by Serving Endpoints and Receiving Endpoints:

Extended audio/visual profile for RTCP based feedback as defined in [8] also called RTP/AVPF.

[ATTRIBUTES]

O	L	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[8]	6R47M

7.5.4.4.2.7 MT RTP Serving Endpoint DLNA Media Format Profile support

7.5.4.4.2.7.1

[GUIDELINE] A Serving Endpoint shall support at least one "RTP support level" as detailed below (in the rest of 7.5.4.4.2.7).

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	Y6R47

NOTE: RTP support levels (A or B) enable better understanding of the extent to which Serving Endpoints support RTP.

7.5.4.4.2.7.2

[GUIDELINE] If a Serving Endpoint claims "RTP support level A" it shall support transmitting at least one media format profile over RTP.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	S5486

NOTE: This is a very basic requirement: an RTP-capable Serving Endpoint must be able to stream at least something over RTP (otherwise it is simply not an RTP-capable Serving Endpoint).

7.5.4.4.2.7.3

[GUIDELINE] If a Serving Endpoint claims "RTP support level B" for each media format profile that it supports transmitting over HTTP, it shall also support transmitting that media format profile over RTP.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	YS548

NOTE: Note that all devices of level B also satisfy level A.

This level of support avoids confusion by the end-user. If a level B Serving Endpoint supports a certain set of media formats and it supports RTP, then RTP is supported for all these media formats.

7.5.4.4.2.8 MT RTP Receiving Endpoint DLNA Media Format Profile support

7.5.4.4.2.8.1

[GUIDELINE] A Receiving Endpoint shall support at least one "RTP support level", as detailed below.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	n/a	MBJYV	

NOTE: RTP support levels (A or B) enable better understanding of the extent to which a Receiving Endpoint supports RTP.

RTP support levels are not signaled between the serving endpoints and the Receiving Endpoints.

7.5.4.4.2.8.2

[GUIDELINE] If a Receiving Endpoint claims "RTP support level A", it shall support receiving at least one media format profile over RTP.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	n/a	7MBJY	

NOTE: This is a very basic requirement: an RTP-capable Receiving Endpoint must be able to receive at least something over RTP (otherwise it is simply not an RTP-capable Receiving Endpoint).

7.5.4.4.2.8.3

[GUIDELINE] If a Receiving Endpoint claims "RTP support level B" for each media format profile that it supports receiving over HTTP, it shall also support receiving that media format profile over RTP.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	n/a	7YFSY	

NOTE: Note that all devices of level B are also satisfying level A.

This level of support avoids confusion by the end-user. If a level B Receiving Endpoint supports a certain set of media formats and it supports RTP, then RTP is supported for all these media formats.

7.5.4.4.2.8.4

[GUIDELINE] If a Receiving Endpoint claims "RTP support level B" and supports a Program Stream based media format profile (over RTP), then it shall also support the corresponding Elementary Stream based media format profile for transport over RTP. In particular:

- If Receiving Endpoint supports MPEG_PS_PAL then it shall also support MPEG_ES_PAL.
- If Receiving Endpoint supports MPEG_PS_NTSC then it shall also support MPEG_ES_NTSC.
- If Receiving Endpoint supports MPEG_PS_PAL_XAC3 then it shall also support MPEG_ES_PAL_XAC3.
- If Receiving Endpoint supports MPEG_PS_NTSC_XAC3 then it shall also support MPEG_ES_NTSC_XAC3.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	n/a	K7YFS	

NOTE: This enables a Serving Endpoint to choose PS or ES encapsulation depending on application requirements.

Note, that the ES based media format profiles are defined for the RTP media transport only (not for HTTP media transport).

7.5.4.4.2.9 MT RTP Payload Type definitions

[GUIDELINE] Serving Endpoints shall use the payload type(s) defined in 7.5.4.4.5.2 A/V Media Format Profiles in the context of RTP transmission when transmitting DLNA media format profile content over RTP,

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	WACXX

NOTE: For each media format profile exactly one RTP encapsulation is defined, except for full TS with zero TTS. This implies that the RTP encapsulation can be inferred from the media format profile as carried in the 4th protocolInfo field except for full TS with zero TTS. Full TS with zero TTS encapsulation can be determined using a=rtpmap in SDP.

Note that Transport Stream and Program Stream based media format profiles are encapsulated as a single RTP stream, whereas other file formats (e.g. MP4, 3GPP) are encapsulated as two separate RTP streams for AV.

7.5.4.4.2.10 MT RTP Header Timestamps during PLAY requests with Speed or Scale Headers

7.5.4.4.2.10.1

[GUIDELINE] A Serving Endpoint shall maintain RTP Header Timestamp values that correspond to playback timing during scaled playback (i.e. a PLAY request with a Scale header with a value not equal to 1) as defined in RFC 2326 Appendix B. The output of the Serving Endpoint during scaled playback shall be compliant with the syntax rules of the Media Format Profile. The values of the RTP Header timestamps shall be compliant with the requirements for the RTP payload format in use (e.g. video/MP2P, video/MP2T, or video/MPV).

For example, a Receiving Endpoint requests a Scale value of 2 for a PLAY request. The Serving Endpoint could comply by dropping every other video frame (or utilize some other algorithm) while maintaining the original frame rate of the content. Assuming a 90 KHz RTP Header timestamp clock timebase and PS encapsulation, the RTP Header timestamp values would increase by 90,000 for each second of playback, even though the NPT time reference would increase by 2.000 each second.

As a further example, if the Scale value was -4 (backwards at 4 times the nominal playback rate), the RTP Header timestamp values will also increase by 90,000 for each second, but the NPT time reference would decrease by -4.000 during the same time period.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[27]	SWACX

NOTE: This guideline clarifies the relationship between the RTP Header Timestamps and the temporal scale of the content being transmitted when the Scale value is not equal to 1. (A Scale value of 1 implies normal playback scale and direction). The output produced by scaling is essentially a "normal" 1X sequence of RTP packets which would have RTP Header Timestamps that are equivalent to a non-scaled content stream. This this is expected be true regardless of the algorithm used by the Serving Endpoint to generate the content.

For example, consider a Serving Endpoint A which implements a Scale value of 2 by dropping every other frame in a 30 fps stream resulting in a content stream which still contains 30 frames per second, but which appears to be playing at twice the normal speed. Next consider Serving Endpoint B which implements the same Scale value of 2 by sending 4 I-Frames a second with a presentation time of 250ms per frame, which will also appear to be playing twice as fast. In both cases the RTP Header Timestamp values would increase by 90,000 each second.

7.5.4.4.2.10.2

[GUIDELINE] A Serving Endpoint shall not change the RTP Header Timestamp values that correspond to playback timing in response to a PLAY request with a Speed Header.

For example, a Receiving Endpoint requests a Speed value of 2 for a PLAY request. The Serving Endpoint would transmit the content at twice the nominal playback rate while maintaining the nominal RTP header timestamp values within each packet.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[27]	TB44G	

NOTE: Note that this only applies to the RTP header timestamp and not to the wall clock as carried within the RTP extension header.

7.5.4.4.2.10.3

[GUIDELINE] If a PLAY request includes both Speed and Scale headers, the Serving Endpoint shall satisfy the parameters of the Scale header, and then deliver the resultant sequence of RTP packets to the network as specified in the Speed header.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[27]	UTB44	

NOTE: This guideline clarifies the order operations if both Scale and Speed headers are included in a PLAY request. Details on the Scale and Speed headers can be found in the subclauses 7.5.4.4.6.2.42 through 7.5.4.4.6.2.45.

7.5.4.4.2.11 MT RTP/RTCP shall use UDP Transport

[GUIDELINE] Serving Endpoints and Receiving Endpoints shall send and receive any RTP and RTCP packets as UDP packets.

[ATTRIBUTES]

M	L	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[38]	B44G2	

7.5.4.4.2.12 MT RTP Unicast support

[GUIDELINE] Serving Endpoints and Receiving Endpoints shall support the transmission of RTP and RTCP packets to unicast IPv4 addresses.

[ATTRIBUTES]

M	L	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	n/a	ACXX8	

7.5.4.4.2.13 MT RTP RTCP Support Required

[GUIDELINE] RTCP shall be implemented as specified in RFC 3550 and as specified by the RTP profile in use with the constraints defined in 7.5.4.4.2 RTP Media Transport: RTP/RTCP Protocols, 7.5.4.4.3 RTP Serving Endpoint Requirements, 7.5.4.4.4 RTP Receiving Endpoint Requirements and 7.5.4.4.6.2 RTP Media Transport: RTSP/SDP Protocols.

[ATTRIBUTES]

M	L	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[38]	CXX84	

NOTE: RTCP is necessary for rate control and synchronization of RTP streams.

7.5.4.4.2.14 MT RTP/RTCP UDP port number usage

[GUIDELINE] RTP Media transport shall be done over an even RTP/UDP port number (i.e. 2n). RTCP control shall be done over the next incremented UDP port number (i.e. 2n+1)

[ATTRIBUTES]

M	L	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[38]	44G2V

7.5.4.4.2.15 MT RTP Unknown RTCP packet types

[GUIDELINE] Serving Endpoints and Receiving Endpoints shall tolerate unknown RTCP packet types.

Tolerate means that the endpoints shall be able to parse and interpret or parse and ignore such packets.

[ATTRIBUTES]

M	C	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[38]	XX84R

7.5.4.4.2.16 MT RTP RTCP Simplified Report Interval Calculation

7.5.4.4.2.16.1

[GUIDELINE] RTCP reports shall be sent at the rate that is in accordance with the SDP provisions (if any).

[ATTRIBUTES]

M	C	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[38]	YFSYN

NOTE: If SDP specifies an RTCP reporting rate, then this rate has the highest precedence. (See 7.5.4.4.6.2.71)

7.5.4.4.2.16.2

[GUIDELINE] In the absence of SDP provisions any RTCP reports shall be generated and sent at the rate that is in accordance to one of the following:

- the RTP profile (AVP or AVPF) in use
- once every 5 seconds randomized within the interval [0.5, 1.5] times, such that the resulting RTCP transmission interval is a random number in the interval [2.5, 7.5]s.

[ATTRIBUTES]

M	C	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[38]	4G2VZ

NOTE: The rules for calculating RTCP report intervals in RFC 3550 are complex to accommodate large multicast sessions.

7.5.4.4.2.17 MT RTP Version Number

[GUIDELINE] Serving Endpoints and Receiving Endpoints shall set the version number in the RTP header to be 2. Serving Endpoints and Receiving endpoints shall accept RTP packets with versions of 2.

[ATTRIBUTES]

M	R	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[38]	SYN4F

7.5.4.4.2.18 MT RTP DLNAQOS RTCP traffic

7.5.4.4.2.18.1

[GUIDELINE] If DLNAQOS as defined 7.2 is implemented, all RTCP messages generated by a Serving Endpoint shall be tagged with DLNAQOS_2, or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3), in accordance Table 11.

[ATTRIBUTES]

M	R	DMS +PU+	M-DMS	n/a	n/a	FSYN4	

NOTE: Because RTP is defined only for UDP, RTCP messages are not subject to the same constraints as HTTP messages sent over TCP. RTCP requests do not have to be the same priority that the server will use to deliver the content.

RTCP packets from a DMS will contain Sender Report (SR) messages when the server is streaming, or Receiver Report (RR) messages when the server is idle. The SR and RR messages are important, but arguably not any more important than the RTP packets because if the network is experiencing congestion, then the RTP stream has more serious problems then any information lost in these messages (e.g. clock sync).

7.5.4.4.2.18.2

[GUIDELINE] If DLNAQOS as defined in 7.2 is implemented, all RTCP messages generated by a Receiving Endpoint shall be tagged with DLNAQOS_3, or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3), in accordance Table 11.

[ATTRIBUTES]

M	R	DMP DMR	M-DMP	MIU	n/a	YV47A	

NOTE: All feedback messages are time critical, and especially important at times when the DLNAQOS_2 RTP traffic is suffering from congestion.

RTCP messages sent by a DMP include Receiver Report (RR) messages and RTCP payload types of 205 and 206. In the RTP/AVPF profile, RTCP payload type 205 is defined as a transport-layer feedback message and 206 is defined as a payload-specific feedback message. This will cover not only RTCP NACK messages, but also other kinds of RTCP-based feedback that we might want to add in the future.

RTCP RR messages contain statistics about lost RTP packets, so they are time critical if the server makes any decisions based on those statistics. Also, all RTCP packets that the client sends must include a RR message. Even if the client just wants to send a NACK, the RTCP packet must also include a RR message.

7.5.4.4.3 RTP Serving Endpoint Requirements

7.5.4.4.3.1 MT RTP timestamp offset

7.5.4.4.3.1.1

[GUIDELINE] All RTP Timestamp values should have an Offset component which is a fixed 32-bit value which is added to the RTP clock value as each RTP Timestamp is created.

[ATTRIBUTES]

S	C	DMS +PU+	M-DMS	n/a	[38]	JYV47	

7.5.4.4.3.1.2

[GUIDELINE] The Serving Endpoint may select an Offset value at the start of the RTP stream, and this value shall be maintained throughout the RTP stream unless a Timestamp Discontinuity is indicated.

[ATTRIBUTES]

O	C	DMS +PU+	M-DMS	n/a	[38]	86GVI	

NOTE: Some RTP payload formats allow a Timestamp Discontinuity to be indicated using the "M" bit in the RTP packet header, or through other means.

7.5.4.4.3.2 MT RTP SSRC uniqueness: Serving Endpoints

7.5.4.4.3.2.1

[GUIDELINE] Serving Endpoints shall ignore SSRC collisions.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[38]	5486G	

NOTE: RTP was designed for large multicast conferencing-type of applications.

SSRC collisions can occur in large conferences where multiple receivers/senders choose their own SSRC.

In a unicast environment, the RTP Serving Endpoint must ignore SSRC collisions and continue the session with the negotiated SSRC values

7.5.4.4.3.3 MT RTP Serving Endpoint RTP retransmission support

7.5.4.4.3.3.1

[GUIDELINE] A Serving Endpoint may retransmit RTP packets if it and the Receiving Endpoint both are using the RTP/AVPF profile.

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	[8]	BJYV4	

7.5.4.4.3.4 MT RTP Serving Endpoint RTP retransmissions

7.5.4.4.3.4.1

[GUIDELINE] A Serving Endpoint that retransmits RTP packets shall use one of the following two retransmission methods:

- The retransmitted packet is formatted using the RTP payload format audio/rtx or video/rtx, in accordance with [10].
- The retransmitted packet is sent as an identical copy of the original packet and is sent to the same UDP port as the original RTP packet.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[10]	486GV	

NOTE: For interoperability reasons it is suggested that the Serving Endpoint implement both transmission methods and choses the one supported by the Receiving Endpoint as announced in 7.5.4.4.6.2.30.

Implementation suggestion: If Serving Endpoint implements a buffering scheme for retransmissions, any encoding rate adaptation techniques, such as bit stream switching, trans-rating and frame skipping, it is recommended not to flush the retransmission buffer, as it can be difficult to reconstruct the original data.

7.5.4.4.3.4.2

[GUIDELINE] A Serving Endpoint that retransmits RTP packets should format the retransmitted packet using the RTP payload format audio/rtx or video/rtx, in accordance with [10].

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	[10]	7MJSW	

7.5.4.4.3.4.3

[GUIDELINE] A Serving Endpoint that retransmits RTP packets may send the retransmitted packet as an identical copy of the original packet and it is sent to the same UDP port as the original RTP packet.

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	[10]	RVS55	

7.5.4.4.3.5 MT RTP Packet Size

7.5.4.4.3.5.1

[GUIDELINE] A Serving Endpoint should limit the size of the RTP packet so that the size of the entire packet including protocol headers will not exceed the Maximum Transmission Unit (MTU) used in the home network.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	[17]	T63U3	

NOTE: The purpose of this recommendation is to avoid IP Packet fragmentation and the performance penalty associated with this

A Serving Endpoint can determine the maximum MTU size for a given connection dynamically using the algorithm described in RFC 1191. If the Serving Endpoint chooses not to, or is unable to determine the size dynamically, it can assume an MTU of size of 1492 bytes is safe.

7.5.4.4.3.5.2

[GUIDELINE] A Serving Endpoint shall not select an RTP packet size in excess of 4096 bytes.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[17]	47MJS	

NOTE: This provides the Receiving Endpoint with an upper limit on the size of an incoming RTP packet.

7.5.4.4.3.6 MT RTP UDP port usage

7.5.4.4.3.6.1

[GUIDELINE] If an RTP media format profile encapsulation (see 7.5.4.4.5.2 A/V Media Format Profiles in the context of RTP transmission) mandates that content be sent as multiple RTP streams, the Serving Endpoint shall support sending them to the same (2n, 2n+1) UDP port pair, even if the Receiving Endpoint requests them to be the same.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[38]	3RVS5	

NOTE: When RTSP is used, the Receiving Endpoint selects the UDP ports for each RTP stream, and although not recommended by [38], it can choose the same UDP port pair for multiple RTP streams.

7.5.4.4.3.6.2

[GUIDELINE] If an RTP media format profile encapsulation (see 7.5.4.4.5.2 A/V Media Format Profiles in the context of RTP transmission) mandates that content be sent as multiple RTP streams, the Serving Endpoint shall support sending them to different (2n, 2n+1) UDP port pairs.

[ATTRIBUTES]

M	R	DMS +PU+	M-DMS	n/a	[38]	R47MJ	

NOTE: This results in each RTP stream being sent on a different RTP session, which is recommended by [38].

7.5.4.4.3.7 MT RTP RTCP First sender report

[GUIDELINE] A Serving Endpoint should transmit a sender report as soon as possible after sending the first RTP packet of an RTP stream.

[ATTRIBUTES]

S	L	DMS +PU+	M-DMS	n/a	[38]	8VQ94	

NOTE: This allows Receiving Endpoints to synchronize RTP streams.

7.5.4.4.3.8 MT RTP Required RTCP SDES items

[GUIDELINE] The only RTCP SDES item required is CNAME.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[38]	VS556	

NOTE: The CNAME item is required as per [38] subclause 6.1.

7.5.4.4.3.9 MT RTP RTCP BYE Packets Recommended

[GUIDELINE] Serving endpoints should send an RTCP BYE packet when leaving an RTP session.

[ATTRIBUTES]

S	L	DMS +PU+	M-DMS	n/a	[38]	Q2KAJ	

7.5.4.4.3.10 MT RTP RTCP Receiver Reports tolerance

[GUIDELINE] Serving Endpoint shall tolerate RTCP Receiver Reports.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[38]	X8VQ9	

7.5.4.4.3.11 MT RTP RTCP Transmission interval in case of RTP translators

[GUIDELINE] If a Serving Endpoint acts as an RTP translator, each received RTCP packet shall be forwarded immediately upon packet arrival. The rule 7.5.4.4.2.16.2 shall not be applied.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[38]	3U3AZ	

NOTE: This rule is compliant to [38] (RTCP processing in translators)

7.5.4.4.3.12 MT RTP Uniqueness of RTP SSRC

[GUIDELINE] Each RTP stream shall use a different SSRC.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[38]	QX8VQ	

NOTE: This is especially helpful to the Receiving Endpoint if multiple RTP streams are sent to the same UDP port.

7.5.4.4.3.13 MT RTP Buffer Fullness Report processing

7.5.4.4.3.13.1

[GUIDELINE] The Serving Endpoint may use Buffer Fullness Reports (BFRs) to compute the average rate of depletion of the Receiving Endpoint's buffer.

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	n/a	63U3A	

NOTE: This information can be used to detect network congestion and transrate the media stream.

7.5.4.4.3.13.2

[GUIDELINE] If BFRs indicate that the Receiving Endpoint's buffer levels are low, then the Serving Endpoint may temporarily increase the transmission rate to fill the Receiving Endpoint's buffer.

Low buffer level is defined as being below the Target Buffer Duration value.

Temporarily increase the transmission rate means that the transmission rate increases until the Serving Endpoint gets a report indicating the buffer level is equal to or greater than the Target Buffer Duration value.

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	n/a	KAJB3	

NOTE: This can be used for faster startup and to make the Receiving Endpoint more tolerant to network jitter.

7.5.4.4.3.13.3

[GUIDELINE] If the Serving Endpoint changes the transmission rate in response to a BFR, RTP timestamps shall not be adjusted to reflect the changed transmission rate.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	43SYB	

7.5.4.4.3.14 MT RTP Transmission rate adaptation

7.5.4.4.3.14.1

[GUIDELINE] If the SETUP request included the Buffer-Info.dlna.org header (guideline 7.5.4.4.6.2.17), and the parameter "BFR=1" was specified on that header, then the Serving Endpoint may perform transmission rate adaptation.

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	n/a	2KAJB	

7.5.4.4.3.14.2

[GUIDELINE] If the Serving Endpoint has indicated that transmission rate adaptation is possible (guideline 7.5.4.4.6.2.80.1) and the Receiving Endpoint specified a Target Buffer Duration for an RTP stream using the Buffer-Info.dlna.org header (guideline 7.5.4.4.6.2.17), then the Serving Endpoint may perform transmission rate adaptation for the first Target Buffer Duration of NPT.

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	n/a	3SYBK

NOTE: Example: If the SETUP request included "TD=5000" on the Buffer-Info.dlna.org header, the Serving Endpoint does not need to pace the data when the transmitting the first 5 seconds worth of data.

7.5.4.4.3.15 MT RTP Wall Clock Time Samples

7.5.4.4.3.15.1

[GUIDELINE] If a Receiving Endpoint requests the Serving Endpoint to add Wall Clock Time samples using the RTSP header WCT.dlna.org, as described in guideline 7.5.4.4.6.2.38.1, then it is strongly recommended for the Serving Endpoint to add Wall Clock Time samples to RTP packets conforming to the guidelines 7.5.4.4.3.16 through 7.5.4.4.3.21 below.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	n/a	SVX64

NOTE: The Wall Clock Time Sample denotes the actual transmission time of the RTP packet very accurately.

This enables Receiving Endpoints to perform clock recovery to enable seamless A/V rendering.

Note that it is up to the Receiving Endpoint whether it chooses to perform clock recovery or not. These guidelines just strongly recommend that the Serving Endpoint does the minimum necessary to make it possible.

For MPEG-2 TS or PS encapsulation, the RTP time stamps provide an alternate means for clock recovery. However, RTP time stamps only denote the actual moment of transmission as accurately as the pacing has been (i.e. 35 ms worst case), whereas the Wall Clock Time Sample is within 2.5 ms accurate. This enables more reliable clock recovery.

For encapsulation of elementary streams the RTP timestamps denote Sample Time and not Transmission Time, making them unsuitable for clock recovery.

7.5.4.4.3.16 MT RTP Wall Clock Time Sample source

7.5.4.4.3.16.1

[GUIDELINE] The Wall Clock Time samples in the RTP packets shall be obtained from the Wall Clock Time source that is used for generating the NTP timestamp in the RTCP SR messages.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[18] [38]	5SVX6

NOTE: The RTCP SR contains a sample of the Wall Clock Time - "NTP timestamp" - when the RTCP packet was sent, along with a sample of the clock generating the RTP timestamps for the RTP stream involved.

This means that the respective RTCP Sender Reports for the different RTP streams (for this media format) can be used to relate all RTP timestamps to the common Wall Clock Time.

This can in turn be used for inter-media synchronization (i.e. "lip sync").

The guidelines 7.5.4.4.3.15 through 7.5.4.4.3.21 effectively enable reconstruction of the serving endpoint Wall Clock Time at the Receiving Endpoint, and thereby serve the same role as e.g. NTP in more conventional RTP implementations.

7.5.4.4.3.16.2

[GUIDELINE] The Wall Clock Time source shall be at least as accurate as specified by the System Clock specification (if any) of the RTP payload content concerned.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	343SY

NOTE: For example, with MPEG-2 content the 90 KHz clock source must be accurate to within ± 30 parts per million (ppm) and have a slew rate of less than 0.075 Hz per second as defined in 13818-1 [5].

7.5.4.4.3.17 MT RTP Wall Clock Time Samples for all packets

[GUIDELINE] If a Serving Endpoint adds Wall Clock Time samples, then they shall be added to each RTP packet of each RTP stream associated with the media format being served.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	Z5SVX	

NOTE: Sampling the Wall Clock Time for each packet of each RTP stream results in the largest number of clock samples per second.

For proper clock reconstruction it is advantageous to have as many samples per second as possible.

7.5.4.4.3.18 MT RTP Wall clock Time Sample accuracy

7.5.4.4.3.18.1

[GUIDELINE] The Wall Clock Time sample in an RTP packet shall represent the moment that the packet is handed over to the network. The standard deviation of the distribution of the errors shall be less than 2.5 milliseconds.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[6]	64JRV	

NOTE: Puts upper bound on additional jitter imposed by the serving endpoint's implementation (i.e. "OS jitter"). Typically, network jitter will dominate over "OS jitter".

The RTP timestamps (Y axis) vs. the actual arrival time on the network (X axis) will be plotted against each other, and a line calculated using a [simple average / least mean squares] fitting algorithm. The slope of the Calculated Line will be unity if the clock source is accurate. A positive or negative slope will indicate a frequency error or "drift". The second derivative or "curve" of the line represents the slew rate of the clock source.

For testing suggestions see footnote[6]

The accuracy of the Wall Clock Time samples will be determined by comparing each Wall Clock Time sample with the actual time the RTP packet is placed on the network as described in [6] over a period of 10 minutes. The slope of the Calculated Line (Figure 34) using a least mean squares will be used to determine accuracy of the Wall clock time samples.

The distribution of the distance between the observed samples and the Calculated Line (Figure 34) must be such that 2 Standard Deviations of the distribution is ± 2.5 milliseconds as shown below in Figure 35

6. This guideline should be tested by connecting a packet capturing device directly to the Device Under Test (DUT) using a crossover Ethernet cable for 802.3 interfaces and/or an 802.11 station or access point in close proximity with no observable radio interference. This will allow the arrival time on the network to be measured as accurately as possible by minimizing the effect of network jitter. However, it should be noted that presence of network jitter does not invalidate the accuracy of this test with respect to clock source accuracy.

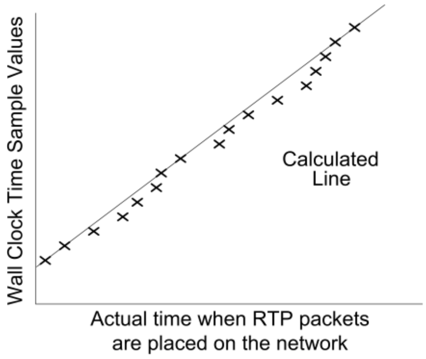


Figure 34 — Calculated Line

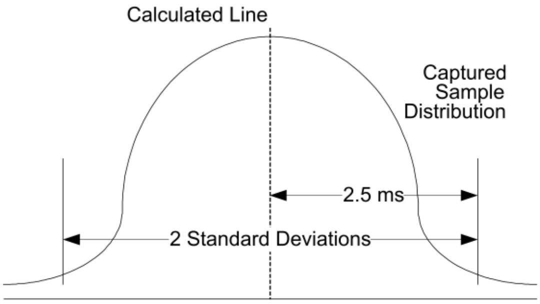


Figure 35 — Wall clock time sample accuracy distribution

7.5.4.4.3.19 MT RTP Wall Clock Time Sample unaffected by Speed, Scale, and BFR.

[GUIDELINE] Wall Clock Time samples shall represent the "Actual" Transmission Time, irrespective of whether the nominal transmission rate is used or not.

Transmission rates other than nominal can occur when: (1) a rate other than 1.0 is requested by the use of the Speed header, or (2) transmission rate adaptation is performed as described in guidelines 7.5.4.4.6.2.80.2 and 7.5.4.4.6.2.80.4.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	n/a	VX64J	

NOTE: The Wall Clock Time Sample will be unaffected by Scale header, Speed header, or BFR as it is not tied to the media stream in any way.

When the transmission rate is increased the Wall Clock Time samples of adjacent packets will have smaller temporal distance compared to normal rate. When the transmission rate is decreased the temporal distances will increase.

[GUIDELINE] Wall Clock Time samples shall represent the "Actual" Transmission Time, irrespective of whether the content is scaled (Scale header not 1) or not.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	n/a	YBKM5	

7.5.4.4.3.20 MT RTP Wall Clock Time Sample representation

[Guideline] The middle 32 bits of the Wall Clock Time sample shall be stored in the RTP packet.

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	[18] [38]	B3TYK	

NOTE: Use of the middle 32 bits reduces packet header overhead and is more commonly used in [38]. It still features a resolution of about 65 KHz (same order of magnitude as 90 KHz RTP timestamps used for PS/TS encapsulation) and a wrap around time of about 18 hours.

From RFC-3550 [38]: Wallclock time (absolute date and time) is represented using the timestamp format of the Network Time Protocol (NTP), which is in seconds relative to 0h UTC on 1 January 1900. The full resolution NTP timestamp is a 64-bit unsigned fixed-point number with the integer part in the first 32 bits and the fractional part in the last 32 bits. In some fields where a more compact representation is appropriate, only the middle 32 bits are used; that is, the low 16 bits of the integer part and the high 16 bits of the fractional part. The high 16 bits of the integer part must be determined independently.

7.5.4.4.3.21 MT RTP Wall Clock Time Sample RTP header extension

7.5.4.4.3.21.1

[Guideline] The 32-bit Wall Clock Time sample in an RTP packet shall be encoded by means of a header extension.

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	n/a	X64JR	

NOTE: Headers extensions offer full backward compatibility: an implementation that does not recognize a header extension must silently ignore it.

7.5.4.4.3.21.2

[Guideline] The X bit in the RTP header shall be set to "1" to indicate the presence of a header extension.

[Attributes]

M	R	DMS +PU+	M-DMS	n/a	[38]	SYBKM	

7.5.4.4.3.21.3

[Guideline] For RTP streams using the RTP/AVP profile the "defined by profile" field in the header extension header shall be set to 0x2356 to uniquely define this DLNA "Wall Clock Time Sample" header extension.

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	n/a	BKM5V	

7.5.4.4.3.21.4

[Guideline] For RTP streams using the RTP/AVPF profile the "defined by profile" field in the header extension header shall be set to 0x2356 to uniquely define this DLNA "Wall Clock Time Sample" header extension.

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	n/a	Q94XX	

7.5.4.4.3.21.5

[GUIDELINE] If no additional header extensions (in addition to Wall Clock Time Sample) are required in the RTP packet header, the "length" field in the header extension header shall be set to 1 to indicate a single 32-bit word in the header extension contents.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	JB3TY	

NOTE: To cater to future needs, optionally, another header extension can follow the Wall Clock Time Sample header extension. In this case the "length" field must indicate the total length of the header extension(s) following the header of the Wall Clock Time Sample header extension. Note that future DLNA guidelines might apply this rule recursively. The case of a single Wall Clock Time Sample header extension is illustrated in Figure 34. The case of another header extension following it is illustrated in Figure 35.

7.5.4.4.3.21.6

[GUIDELINE] Another RTP header extension may follow the Wall Clock Time Sample header extension, as described in [38].

[ATTRIBUTES]

O	R	DMS +PU+	M-DMS	n/a	[38]	AJB3T	

7.5.4.4.3.21.7

[GUIDELINE] If another header extension follows the Wall Clock Time Sample header extension, the "length" field in the Wall Clock Time Sample header extension header shall be set to the total size of the header extension following it (including its header) plus 1 to indicate the single 32-bit word in the Wall Clock Time Sample header extension contents.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	VQ94X	

```
 0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |V=2|P|1|   CC  |M|     PT      |       sequence number         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           timestamp                           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |           synchronization source (SSRC) identifier            |
 +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
 |            contributing source (CSRC) identifiers             |
 |                             ....                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  defined by profile = 0x2356  |           length = 1          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                 Wall Clock Time sample (middle 32 bits)       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 |                      Payload                                  |
 |                                                               |
```

Figure 36 — Packet with Wall Clock Time Sample header extension

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|1|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  defined by profile = 0x2356  |           length = 3          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Wall Clock Time sample (middle 32 bits)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  defined by profile = 0xXXXX  |           length = 1          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Some other 32-bit data                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      Payload                                  |
|                                                               |
```

Figure 37 — Example of packet with another header extension following Wall Clock Time Sample

7.5.4.4.3.22 MT RTP Pacing of RTP Packets

7.5.4.4.3.22.1

[GUIDELINE] In the absence of network congestion, the Serving Endpoint shall be capable of delivering individual RTP packets to the network within 35 milliseconds of a valid "Target Transmission Time". "Target Transmission Time" is defined in guideline 7.5.4.4.3.22.2 below.

The above does not apply when the Serving Endpoint performs transmission rate adaptation as described in guidelines 7.5.4.4.6.2.80.2 and 7.5.4.4.6.2.80.4.

The accuracy of the RTP packet delivery will be determined by capturing the Observed Network Arrival Time as defined in 7.5.4.4.3.18 and comparing it to a valid Target Transmission Time.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	566IS

NOTE: The pace with which RTP packets are delivered to the network must be such that no pre-decoder buffer overflow can occur at the Receiving Endpoint.

This means that RTP packets must be paced in accordance with the standard decoder buffer model of the media format concerned. The standard decoder buffer model determines which are valid delivery times - "Target Transmission Times" - for packets.

For TS/PS encapsulation a valid "Target Transmission Time" is present in the RTP timestamp (See guideline 7.5.4.4.5.4.7).

For ES encapsulation the "Target Transmission Time" is not explicit in the packet, but can be derived from the multiplex of the source material (if present).

Note that 35 milliseconds is a hard limit, no packets are allowed to exceed these bounds.

7.5.4.4.3.22.2

[GUIDELINE] For payload types other than the ones mentioned in 7.5.4.4.5.4.7, the "Target Transmission Time" of an RTP packet shall be defined as the intended delivery time of the first byte of its payload into the pre-decoder buffer of the Receiving Endpoint in a way that is consistent with the standard decoder buffer model applicable to the payload type.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	U3AZW	

7.5.4.4.3.23 MT RTP Maximum reception packet rates

7.5.4.4.3.23.1

[GUIDELINE] Upon reception of the header Max-Prate.dlna.org (see guideline 7.5.4.4.5.4.7), the Serving Endpoint should understand this header and adjust (if necessary) the packet rate based on the max-packet-rate parameter, in order to conform to the maximum packet rate capability of the Receiving Endpoint.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	n/a	94XXR	

NOTE: This rule recommends the Serving Endpoint not to exceed the maximum packet rate specified by the Receiving Endpoint. Otherwise, the Receiving Endpoint might not be able to play all packets

7.5.4.4.3.23.2

[GUIDELINE] A Serving Endpoint that supports the Max-Prate.dlna.org header shall also understand the feature tag dlna.Max-Prate (see guideline 7.5.4.4.6.2.29.2) in a Require header.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[27]	AZWPN	

7.5.4.4.4 RTP Receiving Endpoint Requirements

7.5.4.4.4.1 MT RTP Interpret the P and X bits in the RTP header

[GUIDELINE] Receiving Endpoints shall correctly interpret P bit and the X bit in the RTP packet header.

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[38]	3AZWP	

NOTE: If the P bit is 1, the Receiving Endpoint must remove the padding before processing the packet

If the X bit is 1, the Receiving Endpoint must correctly parse the packet and process any headers that it understands, and tolerate any other headers.

7.5.4.4.4.2 MT RTP Tolerate CSRC fields in the RTP header

[GUIDELINE] Receiving Endpoints shall tolerate contributing sources (CSRCs) in the RTP packet header.

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[38]	5566I	

NOTE: Contributing sources are created only by RTP mixers, which are not required by DLNA.

7.5.4.4.4.3 MT RTP SSRC uniqueness: Receiving Endpoints

[GUIDELINE] Receiving Endpoints shall ignore SSRC collisions.

[ATTRIBUTES]

M	L	DMP DMR	M-DMP	MIU	[38]	MJSWV	

NOTE: RTP was designed for large multicast conferencing-type of applications.

SSRC collisions can occur in large conferences where multiple receivers/senders choose their own SSRC.

In a unicast environment, the RTP Receving Endpoint must ignore SSRC collisions and continue the session with the negotiated SSRC values.

7.5.4.4.4.4 MT RTP Expect Random Starting RTP Sequence Number

[GUIDELINE] A Receiving Endpoint shall accept an arbitrary starting sequence number.

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[38]	S5566	

NOTE: Subclause 5.1 of RFC 3550 ([38]) recommends random starting sequence numbers, so Receiving Endpoints must expect a random starting sequence number.

7.5.4.4.4.5 MT RTP Expect Random Starting RTP Timestamp

[GUIDELINE] A Receiving Endpoint shall accept an arbitrary starting RTP timestamp.

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[38]	JSWVF	

NOTE: Subclause 5.1 of RFC 3550 ([38]) recommends random starting RTP timestamps, so Receiving Endpoints must expect a random starting timestamp.

7.5.4.4.4.6 MT RTP Required RTCP SDES items

[GUIDELINE] The only RTCP SDES item required is CNAME.

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[38]	VIW5Y	

NOTE: The CNAME item is required as per RFC 3550 ([38]) subclause 6.1.

7.5.4.4.4.7 MT RTP Robust handling of RTCP BYE Packet

[GUIDELINE] A Receiving Endpoint shall either accept or gracefully discard RTP data packets received after the reception of the RTCP BYE packet

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[38]	6GVIW	

NOTE: BYE packets can be received before or after the transmission of RTP packets has actually ended because of network reordering or retransmission.

7.5.4.4.4.8 MT RTP Unknown RTP extensions

[GUIDELINE] Receiving Endpoints shall tolerate RTP header extensions they do not support.

Tolerate means that the Receiving Endpoint is able to "parse and interpret" or "parse and ignore" header extensions.

[ATTRIBUTES]

M	R	DMP DMR	M-DMP	MIU	[38]	SWVFR	

7.5.4.4.4.9 MT RTP Out of order RTP packets and jitter conditions

7.5.4.4.4.9.1

[GUIDELINE] Receiving Endpoints receiving RTP packets shall accept out-of-order packets.

Shall accept means that the Receiving Endpoint is able to "receive and reorder" or "receive and ignore" header extensions.

[ATTRIBUTES]

M	R	DMP DMR	M-DMP	MIU	[38]	47AI8	

7.5.4.4.4.9.2

[GUIDELINE] A Receiving Endpoint should be capable of processing RTP packets that arrive with up to 200ms of total jitter.

[ATTRIBUTES]

S	C	DMP DMR	M-DMP	MIU	[38]	GVIW5	

NOTE: This jitter is the sum of the OS jitter (which is 35 ms) plus the network induced jitter.

7.5.4.4.4.9.3

[GUIDELINE] If a Receiving Endpoint sends Buffer Fullness Reports, it shall tolerate variable rate transmission of RTP packets for a period up to the duration of the Receiving Endpoint's buffer.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[38]	YN4FN	

7.5.4.4.4.10 MT RTP/AVPF support

[GUIDELINE] A Receiving Endpoint may use the RTP/AVPF profile, and send RTCP-based feedback messages of the types that Serving Endpoint has indicated are acceptable.

[ATTRIBUTES]

O	C	DMP DMR	M-DMP	MIU	[8]	V47AI	

NOTE: The Serving Endpoint uses SDP to indicate if RTP/AVPF is used, and which feedback message types are acceptable.

7.5.4.4.4.11 MT RTP Retransmission requests

[GUIDELINE] If the RTP/AVPF profile is used, and if the Serving Endpoint has indicated that RTCP nack feedback messages are acceptable, then the Receiving Endpoint shall use RTCP nack feedback message to request any desired RTP packet retransmissions.

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[8]	7AI8E	

7.5.4.4.4.12 MT RTP audio/rtx and video/rtx support

7.5.4.4.4.12.1

[Guideline] A Receiving Endpoint which requests retransmission of RTP packets should support the audio/rtx and video/rtx RTP payload formats for retransmitted packets.

[Attributes]

S	A	DMP DMR	M-DMP	MIU	[10]	4FNJE	

7.5.4.4.4.12.2

[Guideline] A Serving Endpoint that supports the audio/rtx and video/rtx RTP payload formats for retransmitted packets shall use the SSRC-multiplexing method described in clause 5 of [10] and shall not use the session-multiplexing method (also described in clause 5 of [10]).

[Attributes]

M	L	DMS +PU+	M-DMS	n/a	[10]	84R9I	

7.5.4.4.4.13 MT RTP packets that are retransmitted as identical copies

[Guideline] A Receiving Endpoint which requests retransmission of RTP packets may support that RTP packets are retransmitted as identical copies of the original RTP packets.

[Attributes]

O	A	DMP DMR	M-DMP	MIU	n/a	N4FNJ	

7.5.4.4.4.14 MT RTP Rules for counting RTP packets when retransmission requests are supported

7.5.4.4.4.14.1

[Guideline] A Receiving Endpoint which requests the retransmission of an RTP packet using an RTCP nack feedback message, shall count that packet as a lost packet in RTCP Receiver Reports, even if the RTP packet is subsequently received.

[Attributes]

M	A	DMP DMR	M-DMP	MIU	n/a	X84R9	

7.5.4.4.4.14.2

[Guideline] A Receiving Endpoint which requests the retransmission of an RTP packet using an RTCP nack feedback message, shall not include that packet when computing the value for the "interarrival jitter" field in RTCP Receiver Reports, even if the RTP packet is subsequently received.

[Attributes]

M	A	DMP DMR	M-DMP	MIU	n/a	4R9IZ	

7.5.4.4.4.15 MT RTP Buffer Fullness Reports

7.5.4.4.4.15.1

[Guideline] If the RTP/AVPF profile is used, and if the Serving Endpoint has indicated that RTCP bfr feedback messages are acceptable, then the Receiving Endpoint should include Buffer Fullness Reports (BFR's) with all RTCP Receiver Reports that it sends.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[8]	VZTX5

NOTE: BFR's can be used to detect network congestion and ensure the RENEDERING ENDPOINTS jitter buffer remains full.

7.5.4.4.4.15.2

[GUIDELINE] A Receiving Endpoint that sends BFR's shall send them at the rate that is in accordance with SDP provisions (if any).

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[41]	G2VZT

NOTE: See guideline 7.5.4.4.6.2.76.2 for SDP provisions that define the rate at which BFRs must be sent. The Serving Endpoint requires BFR's to be sent at this rate to detect congestion.

7.5.4.4.4.15.3

[GUIDELINE] A Receiving Endpoint that sends BFRs shall use the following syntax:

- An RTCP feedback message as defined in RTP/AVPF [8] subclause 6.1 shall be used
- The FMT field shall be set to 3 indicating the bfr feedback message type
- The PT field shall be set to 205 indicating a transport layer feedback message
- The feedback control information (FCI) section consists of ten fields. The fields shall occur in the FCI in the order listed below.
- N1: (32 bits.) Shall be set to the number of free bytes in the Receiving Endpoint's network de-jitter buffer
- N2: (32 bits.) Shall be set to the current size, in bytes, of the Receiving Endpoint's network de-jitter buffer. The current size is the current maximum size and not the current utilization or amount of data in the buffer.
- N3: (16 bits.) Shall be set to the amount of data queued in the Receiving Endpoint's network de-jitter buffer counted in milliseconds. If this information is not available, this field shall be set to 0xFFFF. If the amount of data is more than 65534 milliseconds, the fields shall be set to 0xFFFE.
- Playout Delay: (16 bits.) Shall be set to the difference between the scheduled playout time of the next ADU (Application Data Unit) to be transferred to the coded data buffer or decoded if coded data buffer is not presented and the time of sending the BFR, as measured by the media playout clock, expressed in milliseconds. If this information is not available, the Receiving Endpoint shall set this value to 0xFFFF. In case of an empty buffer, the playout delay is not defined and the field shall be set to 0xFFFF
- NSN: (16 bits.) Shall be set to the RTP sequence number of the next ADU to be transferred or decoded for the SSRC reported on. In the case where the buffer does not contain any packets for this SSRC, the next not yet received RTP sequence number shall be reported, i.e. an NSN value that is one larger than the least significant 16 bits of the RTCP SR or RR report block's "extended highest sequence number received".
- NUN: (16 bits.) Shall be set to the unit number (within the RTP packet) of the next ADU to be transferred or decoded as defined in the payload format subclause of these guidelines, 7.5.4.4.5.2 A/V Media Format Profiles in the context of RTP transmission, for the given RTP payload format. The first unit in a packet has a unit number equal to zero. The unit number is incremented by one for each ADU in an RTP packet. In the case of RTP payload formats where each packet carries a single ADU, or where the ADU is not defined, the NUN field shall be set to zero.
- D1: (32 bits.) Shall be set to the number of free bytes in the Receiving Endpoint's coded data buffer, or 0 if the pre-decoder buffer is combined with the network de-jitter buffer.

The field shall be set to 0 if the RTSP Buffer-Info.dlna.org header did not provide a size for the coded data buffer.

- D2: (32 bits.) Shall indicate the current size, in bytes, of the Receiving Endpoint's coded data buffer, or 0 if the pre-decoder buffer is combined with the network de-jitter buffer. The field shall be set to 0 if the RTSP Buffer-Info.dlna.org header did not provide a size for the coded data buffer.
- D3: (16 bits.) Shall be set to the amount of data queued in the Receiving Endpoint's coded data buffer counted in milliseconds. If this information is not available, this field shall be set to 0xFFFF. If the amount of data is more than 65534 milliseconds, the fields shall be set to 0xFFFE.
- TD: (16 bits.) Shall be set to the current value of Target Buffer Duration, as defined in guideline 7.5.4.4.6.2.17. The field shall be set to 0 if the Buffer-Info.dlna.org header was not included in the RTSP SETUP request, or if the "TD" parameter on that header was not present.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[8]	ZTX53	

NOTE: The fields N2 and D2 are necessary because the Receiving Endpoint can decide to increase the sizes of these buffers as a result of headers in the PLAY response, or as a result of network congestion, for example.

See Figure 37 for more information.

The definition of Application Data Unit is specific to each RTP payload format. See definitions in 7.5.4.4.5.2 A/V Media Format Profiles in the context of RTP transmission, and the comments about the NUN field, below.

In the two-buffer model where no RTP header is available in the coded data buffer, the reported NSN and the associated playout delay and NUN must use the next to be transferred RTP packet's sequence number in the transmission (network) jitter buffer.

For the NUN field:

For example in the case of H.264 (AVC), an ADU is defined as a NAL unit and for audio profiles, an ADU is an audio frame.

In the case of RTP payload formats where each packet carries a single ADU (for example H.263 and MPEG-4 Visual Simple Profile) the NUN field will be set to zero.

MPEG-2 PS streams consist of a stream of bytes without payload-level framing. Therefore NUN must always be 0 for such streams.

Future additions of media encoding or transports capable of having more than one ADU in each RTP payload shall define what shall be counted as an ADU for this format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|   FMT=3  |     PT=205     |          length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of packet sender                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of media source                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    N1 (free NDB space)                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    N2 (size of NDB)                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  N3 (NDB level in ms)           |      PD (Playout Delay)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  NSN (next sequence number)     |  NUN (next ADU's unit number) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    D1 (free CDB space)                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    D2 (size of CDB)                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  D3 (CDB level in ms)           |  TD (target buffer duration)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   NDB: network de-jitter buffer
   CDB: coded data buffer
```


Figure 38 — BFR packet format

7.5.4.4.4.16 MT RTP Tolerate Wall Clock Time Sample RTP header extension

7.5.4.4.4.16.1

[GUIDELINE] Receiving Endpoint shall tolerate the Wall Clock Time sample RTP header extension as defined in guidelines 7.5.4.4.3.15 through 7.5.4.4.3.21.

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[38]	5398X

NOTE: RFC-3550 [38] already mandates RTP header extensions must be silently ignored if they cannot be interpreted.

The aim of the referenced guidelines is for a Receiving Endpoint to actively use the Wall Clock Time samples to perform clock recovery. As stated before, this is not mandatory.

7.5.4.4.5 RTP Media Transport: Adaptation of Media Format Profiles

7.5.4.4.5.1 General

Reference [56] specifies audio, video, and encapsulation characteristics for media content that conforms to the list of DLNA mandatory and optional media format profiles when the transport protocol is HTTP. 7.5.4.4.5.2 A/V Media Format Profiles in the context of RTP transmission uses Reference [56] as the basis, and defines additional constraints and requirements necessary to reuse the Profile ID values and exchange content using RTP. This subclause describes an adaptation layer or filtering layer to tailor the media format profiles defined in [56] for usage over RTP.

7.5.4.4.5.2 A/V Media Format Profiles in the context of RTP transmission

7.5.4.4.5.2.1 MT RTP Profile ID usage

[GUIDELINE] For transport over RTP, A/V content shall use the same Profile ID values defined in [56].

[ATTRIBUTES]

M	C	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56]	IZ5XA

7.5.4.4.5.2.2 MT RTP MPEG-2 media format profiles

7.5.4.4.5.2.2.1

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- MPEG-2 video with any type of companion audio
- Full Single Program Transport Stream encapsulation
- DLNA Transport Packets without Timestamp fields (188 byte ISO profiles)

The RTP encapsulation shall use either:

- The payload format MP2T as defined in [25] and [40] with constraints in guideline 7.5.4.4.5.4.2, or
- The payload format vnd.dlna.mpeg-mp2t as defined in guideline 7.5.4.4.5.4.3

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56] [25]	NJEQX

NOTE: Full Single Program Transport Streams with zero TTS carrying MPEG-2 media format profiles.

7.5.4.4.5.2.2.2

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- MPEG-2 video with any type of companion audio
- Partial Single Program Transport Stream encapsulation
- DLNA Transport Packets without Timestamp fields (188 byte ISO profiles)

The RTP encapsulation shall use payload format vnd.dlna.mpeg-mp2t as defined in guideline 7.5.4.4.5.4.3.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56]	AI8EX

NOTE: Partial Single Program Transport Streams with zero TTS carrying MPEG-2 media format profiles.

7.5.4.4.5.2.2.3

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- MPEG-2 video with any type of companion audio
- Partial or Full Single Program Transport Stream encapsulation
- Non-zero TTS

The RTP encapsulation shall use payload format vnd.dlna.mpeg-tts as defined in guideline 7.5.4.4.5.4.4.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56]	X5398

NOTE: Partial or Full Single Program Transport Streams with non-zero TTS carrying MPEG-2 media format profiles.

7.5.4.4.5.2.2.4

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- MPEG-2 video with any type of companion audio
- Program Stream encapsulation

The RTP encapsulation shall use payload format MP2P as defined in [25] and [40].

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56] [25] [40]	9IZ5X

NOTE: Program Streams carrying MPEG-2 media format profiles.

7.5.4.4.5.2.2.5

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- MPEG-2 video with any type of companion audio
- MPEG-2 Elementary Stream encapsulation [25]

The RTP encapsulation for video shall use payload format MPV as defined in [25] and the RTP encapsulation for audio shall comply with 7.5.4.4.5.3 Adaptation of Audio-only Media Format Profiles to RTP, and Adaptation of the Audio component of A/V Media Format Profiles when transmitted as separate elementary streams.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56] [25]	FNJEQ

NOTE: MPEG-2 Elementary Streams carrying MPEG-2 media format profiles.

7.5.4.4.5.2.3 MT RTP AVC media format profiles

7.5.4.4.5.2.3.1

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- AVC video with any type of companion audio
- Full Single Program Transport Stream encapsulation
- DLNA Transport Packets without Timestamp fields (188 byte ISO profiles)

The RTP encapsulation shall use either:[25]

- The payload format MP2T as defined in [25] and [40] with constraints in guideline 7.5.4.4.5.4.2, or
- The payload format vnd.dlna.mpeg-mp2t as defined in guideline 7.5.4.4.5.4.3.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56] [25]	TX539

NOTE: Full Single Program Transport Streams with zero TTS carrying AVC media format profiles.

7.5.4.4.5.2.3.2

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- AVC video with any type of companion audio
- Partial Single Program Transport Stream encapsulation
- DLNA Transport Packets without Timestamp fields (188 byte ISO profiles)

The RTP encapsulation shall use payload format vnd.dlna.mpeg-mp2t as defined in guideline 7.5.4.4.5.4.3 with the following additional constraints. The TS packet size shall be 188 bytes and the 4-byte TTS defined in [56] shall be excluded.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56]	R9IZ5

NOTE: Partial Single Program Transport Streams with zero TTS carrying AVC media format profiles.

7.5.4.4.5.2.3.3

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- AVC video with any type of companion audio
- Partial or Full Single Program Transport Stream encapsulation
- Non-zero TTS

The RTP encapsulation shall use payload format vnd.dlna.mpeg-tts as defined in guideline 7.5.4.4.5.4.4.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56]	Z5XAW

NOTE: Partial or Full Single Program Transport Streams with non-zero TTS carrying AVC media format profiles.

7.5.4.4.5.2.3.4

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- AVC video with any type of companion audio
- Profile ID indicating MP4 or 3GPP encapsulation

When these Profile ID values are used in conjunction with RTP all of the following shall apply.

- Audio and Video shall be encapsulated as elementary streams (no use of the MPEG-4 file format or 3GPP file format)
- The RTP encapsulation for video shall use payload format H264 [43] with constraints in 7.5.4.4.5.6 Guidelines for the encapsulation for AVC elementary streams.

- The RTP encapsulation for audio shall comply with 7.5.4.4.5.3 Adaptation of Audio-only Media Format Profiles to RTP, and Adaptation of the Audio component of A/V Media Format Profiles when transmitted as separate elementary streams.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56] [43]	JEQXQ

NOTE: AVC media format profiles that indicate MP4 or 3GPP encapsulation

7.5.4.4.5.2.4 MT RTP MPEG-4 Part 2 media format profiles

7.5.4.4.5.2.4.1

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- MPEG-4 Part 2 video with any type of companion audio
- Full Single Program Transport Stream encapsulation
- DLNA Transport Packets without Timestamp fields (188 byte ISO profiles)

The RTP encapsulation shall use either:

- The Payload format MP2T ([25], [40]) with constraints in 7.5.4.4.5.4.2, or
- The Payload format vnd.dlna.mpeg-mp2t as defined in guideline 7.5.4.4.5.4.3

.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56] [25]	I8EXF

NOTE: Full Single Program Transport Streams with zero TTS carrying MPEG-4 Part 2 media format profiles.

7.5.4.4.5.2.4.2

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- MPEG-4 Part 2 video with any type of companion audio
- Partial Single Program Transport Stream encapsulation
- DLNA Transport Packets without Timestamp fields (188 byte ISO profiles)

The RTP encapsulation shall use payload format vnd.dlna.mpeg-mp2t as defined in guideline 7.5.4.4.5.4.3.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56]	IW5Y5

NOTE: Partial Single Program Transport Streams with zero TTS carrying MPEG-4 Part 2 media format profiles.

7.5.4.4.5.2.4.3

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- MPEG-4 Part 2 video with any type of companion audio
- Partial or Full Single Program Transport Stream encapsulation
- Non-zero TTS

The RTP encapsulation shall use payload format vnd.dlna.mpeg-tts as defined in guideline 7.5.4.4.5.4.4 with the following additional constraints. The TS packet size shall be 192 bytes and the 4-byte TTS defined in [56] shall be included.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56]	EQXQ6

NOTE: Partial or Full Single Program Transport Streams with non-zero TTS carrying MPEG-4 Part 2 media format profiles.

7.5.4.4.5.2.4.4

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- MPEG-4 Part 2 video with any type of companion audio
- Profile ID indicating MP4 or ASF encapsulation

When these Profile ID values are used in conjunction with RTP all of the following shall apply:

- Audio and Video shall be encapsulated as elementary streams (no use of the MPEG-4 or ASF file format)
- The RTP encapsulation for video shall use payload format mpeg4-generic [42] with constraints in 7.5.4.4.5.7 Guidelines for the encapsulation for MPEG-4 part 2 elementary streams.
- The RTP encapsulation for audio shall comply with 7.5.4.4.5.3 Adaptation of Audio-only Media Format Profiles to RTP, and Adaptation of the Audio component of A/V Media Format Profiles when transmitted as separate elementary streams

.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56] [42]	8EXFY

NOTE: MPEG-4 Part 2 media format profiles that indicate MP4 or ASF encapsulation.

7.5.4.4.5.2.5 MT RTP H.263 media format profiles

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- H.263 video with any type of companion audio
- Profile ID indicating MP4 encapsulation

When these Profile ID values are used in conjunction with RTP all of the following shall apply:

- Audio and Video shall be encapsulated as elementary streams (no use of the MPEG-4 file format)
- The RTP encapsulation for video shall use payload format H263-2000 [40], [30] with constraints in 7.5.4.4.5.9 Guidelines for the encapsulation of H.263 streams.
- The RTP encapsulation for audio shall comply with 7.5.4.4.5.3 Adaptation of Audio-only Media Format Profiles to RTP, and Adaptation of the Audio component of A/V Media Format Profiles when transmitted as separate elementary streams.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[40], [56], [30]	W5Y5T

NOTE: H.263 media format profiles that indicate MP4 encapsulation

7.5.4.4.5.2.6 MT RTP WMV media format profiles

[GUIDELINE] For A/V Profile ID values in [56] for which the following holds:

- WMV video with any type of companion audio
- Media format profiles indicating use of ASF encapsulation

When these Profile ID values are used in conjunction with RTP all of the following shall apply:

- Audio and Video shall be encapsulated as elementary streams (no use of the ASF file format)
- The RTP encapsulation for video shall use payload format WMV [92] with constraints in guideline 7.5.4.4.5.5 Guidelines for encapsulation of WMA and WMV elementary streams.
- The RTP encapsulation for audio shall comply with 7.5.4.4.5.3 Adaptation of Audio-only Media Format Profiles to RTP, and Adaptation of the Audio component of A/V Media Format Profiles when transmitted as separate elementary streams.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56], [92]	Y5T9R

NOTE: WMV media format profiles.

7.5.4.4.5.3 Adaptation of Audio-only Media Format Profiles to RTP, and Adaptation of the Audio component of A/V Media Format Profiles when transmitted as separate elementary streams

7.5.4.4.5.3.1 MT RTP Media Format Profile ID usage

[GUIDELINE] For transport over RTP, Audio-only content shall use the same Profile ID values defined in [56].

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56]	EXFYZ

7.5.4.4.5.3.2 MT RTP Audio Elementary Streams of A/V media format profiles

[GUIDELINE] Audio components of A/V profiles in [56] transmitted over RTP as separate elementary streams shall follow the adaptation guidelines defined in 7.5.4.4.5.3 Adaptation of Audio-only Media Format Profiles to RTP, and Adaptation of the Audio component of A/V Media Format Profiles when transmitted as separate elementary streams.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56]	5Y5T9

7.5.4.4.5.3.3 MT RTP Elementary Streams of Audio-only media format profiles

[GUIDELINE] Audio content of Audio-only profiles in [56] shall be transmitted over RTP as elementary streams using the adaptation guidelines defined in 7.5.4.4.5.3.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[56]	RXG3B

7.5.4.4.5.3.4 MT RTP LPCM media format profiles

[GUIDELINE] The RTP encapsulation for LPCM shall use payload format L16 [39].

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[39]	ISH26

7.5.4.4.5.3.5 MT RTP MP3,MP3X, MPEG-2 L2 media format profiles

[GUIDELINE] The RTP encapsulation for MP3, MP3X, and MPEG-2 L2 shall use payload format MPA [25].

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[25]	WPN9Z

7.5.4.4.5.3.6 MT RTP AC3, XAC3 media format profiles

[GUIDELINE] The RTP encapsulation for AC-3 and XAC3 shall use payload format ac3 [9].

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[93] [9]	4XXRY

7.5.4.4.5.3.7 MT RTP AAC, HE-AAC media format profiles

[GUIDELINE] The RTP encapsulation for AAC and HE-AAC shall use payload format mpeg4-generic [42] with constraints in 7.5.4.4.5.8 Guidelines for the encapsulation of MPEG-4 AAC streams.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[42]	FRXG3

7.5.4.4.5.3.8 MT RTP G726 media format profiles

[GUIDELINE] The RTP encapsulation for G726 shall use payload format G726-32 [39].

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[39]	6ISH2

7.5.4.4.5.3.9 MT RTP WMA media format profiles

[GUIDELINE] The RTP encapsulation for WMA shall use payload format WMA [92] with constraints in 7.5.4.4.5.5 Guidelines for encapsulation of WMA and WMV elementary streams.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[92]	ZWPN9

7.5.4.4.5.3.10 MT RTP AMR media format profiles

[GUIDELINE] The RTP encapsulation for AMR shall use payload format AMR [36] with constraints in 7.5.4.4.5.10 Guidelines for the encapsulation AMR streams.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[36] [56]	VFRXG

7.5.4.4.5.3.11 MT RTP AMR- WBPlus media format profiles

[GUIDELINE] The RTP encapsulation for AMR-WBplus shall use payload format AMR-WB+ [31] with constraints in 7.5.4.4.5.11 Guidelines for the encapsulation AMR-WBplus streams.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[31] [56]	66ISH

7.5.4.4.5.4 Guidelines for encapsulation of MPEG-2 streams

7.5.4.4.5.4.1 MT RTP Payload Formats for MPEG TS encapsulated content

7.5.4.4.5.4.1.1

[GUIDELINE] The guidelines in 7.5.4.4.5.4.1 apply to guidelines 7.5.4.4.5.4.2, 7.5.4.4.5.4.3 and 7.5.4.4.5.4.4.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[5] [25]	WVFRX

7.5.4.4.5.4.1.2

[GUIDELINE] All RTP Timestamps for MPEG TS encapsulated content will be in 90 kHz clock units.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[25]	SH26Q

7.5.4.4.5.4.1.3

[GUIDELINE] The 90 kHz clock utilized as the basis for the RTP Header Timestamp field shall be synchronized with the MPEG System Clock of the content.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[5] [25]	PN9ZA

7.5.4.4.5.4.1.4

[GUIDELINE] The RTP Timestamp values may differ from the MPEG program_clock_reference_base value in the PCR field.

[ATTRIBUTES]

O	C	DMS +PU+	M-DMS	n/a	[25]	XXRYP

7.5.4.4.5.4.1.5

[GUIDELINE] An ADU (Application Data Unit) shall be one complete TS packet as specified in the guidelines for the RTP payload format in use.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[25]	3TYKK

NOTE: The size of a TS packet differs depending on the RTP payload format in use.

7.5.4.4.5.4.2 MT RTP MP2T RTP Payload Format Definition

7.5.4.4.5.4.2.1

[GUIDELINE] The MP2T RTP Payload format can only be used to transfer Full SPTS streams in RTP. If this RTP Payload format Definition is used, the RTP packets shall meet the following guidelines.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[25]	N9ZAL

NOTE: This RTP payload format definition is a strict definition of RFC2250 using Full SPTS streams.

7.5.4.4.5.4.2.2

[GUIDELINE] Each MPEG TS Packet shall be 188 bytes in length and conform to the requirements of 13818-1 [5] . The MP2T Payload format can only be utilized with DLNA Media Format Profiles that utilize MPEG2 TS encapsulation and have 188 byte TS Packets. Specifically MP2T is not a valid RTP payload format for DLNA media format profiles utilizing the 192 byte TS packet syntax

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[5]	XRYPY

7.5.4.4.5.4.2.3

[GUIDELINE] The RTP Timestamp shall represent the ideal transmission time of the first byte of the first MPEG TS Packet contained within the RTP Payload as defined in RFC2250 and further clarified below.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[25]	KM5VI

7.5.4.4.5.4.2.4

[GUIDELINE] The ideal transmission time of the first byte of an MPEG2 TS Packet shall be calculated as follows:

$TSTimestampY = Offset + 90KHZX + ((\Delta T_{PCR} / N) * I)$

where:

I = Index value of the TS Packet relative to the last TS Packet containing a PCR value. The first TS Packet after a PCR value will have an Index of 1, the second TS Packet an Index of 2, and so on.

N = Number of TS Packets between TS Packets with PCR values plus one for the TS Packet with the PCR value. For example, if a SPTS has 2,000 TS Packets per second, and PCR values occur every 100 ms, the value of N would be 200.

90KHZX = The value of the 90 KHz clock associated with the last TS Packet containing a PCR value.

ΔT_{PCR} = Delta time value between PCR values as defined in 3.2.

Offset = RTP Timestamp Offset as defined in 7.5.4.4.3.1.

[Attributes]

M	C	DMS +PU+	M-DMS	n/a	n/a	YKK6T	

7.5.4.4.5.4.2.5

[Guideline] The Serving Endpoint shall utilize either a static Payload ID of '33' or a dynamic Payload Format of 'MP2T' in the SDP Media Description Fields as defined in 0 NOTE: The 4th_field is the 4th field of the protocolInfo value supplied by the serving endpoint for this content binary as defined in guideline 7.4.1.3.16.

MT RTP SDP Media Description Fields.

[Attributes]

M	C	DMS +PU+	M-DMS	n/a	n/a	M5VI8	

7.5.4.4.5.4.3 MT RTP vnd.dlna.mpeg-mp2t RTP Payload format Definition

7.5.4.4.5.4.3.1

[Guideline] The vnd.dlna.mpeg-mp2t RTP Payload format definition can be used to transfer Partial or Full SPTS streams in RTP. If this TS Payload Definition is used, the RTP packets shall meet the following guidelines 7.5.4.4.5.4.3.2, 7.5.4.4.5.4.3.3 and 7.5.4.4.5.4.3.4.

[Attributes]

M	L	DMS +PU+	M-DMS	n/a	n/a	4JRVZ	

NOTE: This RTP TS Payload format Definition is a looser definition of RFC2250 MP2T that allows both Partial and Full SPTS streams. The RTP Timestamp calculations are identical even though the temporal relationship between any two adjacent TS Packets is not always the same.

7.5.4.4.5.4.3.2

[Guideline] This payload format definition is identical to MP2T as defined in guideline 7.5.4.4.5.4.2.2, except for the name and that it can also be used to carry partial SPTS streams.

[Attributes]

M	L	DMS +PU+	M-DMS	n/a	n/a	RYPY6	

7.5.4.4.5.4.3.3

[Guideline] The RTP Packet Timestamp shall be calculated in the same manner as described in 7.5.4.4.5.4.2.3 and 7.5.4.4.5.4.2.4 for the MP2T Payload format Definition.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	n/a	KK6T3	

7.5.4.4.5.4.3.4

[GUIDELINE] The Serving Endpoint shall utilize a dynamic Payload Format of 'vnd.dlna.mpeg-mp2t' in the SDP Media Description Fields as defined in 0.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	n/a	5VI84	

7.5.4.4.5.4.3.5

[GUIDELINE] To reduce jitter on the transmission of PCR timestamps, the Serving Endpoint should send RTP packets as soon as it contains a TS packet with a PCR, without waiting for the max payload size to be reached.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	n/a	JRVZQ	

7.5.4.4.5.4.4 MT RTP vnd.dlna.mpeg-tts RTP Payload format Definition

7.5.4.4.5.4.4.1

[GUIDELINE] The vnd.dlna.mpeg-tts RTP Payload format definition can be used to transfer Partial or Full SPTS streams in RTP. If this RTP Payload format definition is used, the RTP packets shall meet guidelines 7.5.4.4.5.4.4.2, 7.5.4.4.5.4.4.7 and 7.5.4.4.5.4.4.8.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	VI84D	

NOTE: This RTP Payload format Definition is designed to preserve individual TS Packet timing references in a Partial SPTS stream.

7.5.4.4.5.4.4.2

[GUIDELINE] The vnd.dlna.mpeg-tts RTP Payload format definition can only be utilized with DLNA Media Format Profiles that utilize MPEG2 TS encapsulation and have 192 byte TS Packets. It shall be used for DLNA Media Format Profiles utilizing the DLNA-defined 192 byte TS packet format with timestamp.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[5]	RVZQ5	

NOTE: The individual timestamps in each TS Packet are designed to preserve the temporal relationship between each TS packet and its two adjacent packets.

7.5.4.4.5.4.4.3

[GUIDELINE] Each TS Packet shall be 192 bytes in length consisting of a 4 byte TTS Timestamp field followed by a 188 byte MPEG TS packet conforming to the requirements of 13818-1 [5].

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[5]	VZQ5R	

7.5.4.4.5.4.4.4

[GUIDELINE] The TTS Timestamp field shall be in 27 MHz clock units. The 27 MHz clock utilized as the basis for the TTS Timestamps shall be synchronized with the MPEG System Clock of the content.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[5]	RMVJ9

7.5.4.4.5.4.4.5

[GUIDELINE] The TTS Timestamp value is a 32-bit binary counter and is not defined using the same syntax as the PCR field. If this TS Definition is utilized, the Serving Endpoint shall ensure that the individual TTS Timestamp values have an accuracy of ± 500 ns.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[5]	Q5RMV

7.5.4.4.5.4.4.6

[GUIDELINE] The TTS Timestamp field shall contain valid timestamp values.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[5]	5RMVJ

NOTE: Note that this is different from how the TTS Timestamps are used in HTTP.

7.5.4.4.5.4.4.7

[GUIDELINE] The RTP Timestamp shall be derived from the TTS Timestamp of the first TS Packet in the RTP Payload using the following equations:

For the first RTP Timestamp (TSRTP) in the RTP stream, or after a Program System Clock discontinuity:

TSRTP = Offset + (TSTTS / 300)

where Offset is the RTP Offset defined in 7.5.4.4.5.4.1 and TSTTS is the 27MHz TTS 32-bit Timestamp.

For subsequent RTP Timestamps:

$TSRTP_N = TSRTP_{N-1} + ((TSTTS_N - TSTTS_{N-1}) / 300)$

Where $TSRTP_N$ is the RTP time stamp of the Nth RTP packet.

Where $TSTTS_N$ is the TTS time stamp of the first TS packet in the payload of the Nth RTP packet.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	ZQ5RM

NOTE: These equations compensate for the difference in time base between the TTS Timestamp (27 MHz) and the RTP Timestamp (90 KHz). It allows the RTP Timestamp to reach its full value (roll over at 232). Implementers must account for roller in the $TSTTS_N - TSTTS_{N-1}$ expression.

7.5.4.4.5.4.4.8

[GUIDELINE] The Serving Endpoint shall utilize a dynamic Payload Format of 'vnd.dlna.mpeg-tts' in the SDP Media Description Fields as defined in 0.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	n/a	I84D8	

7.5.4.4.5.4.5 MT RTP payload format support for TS without TTS

[GUIDELINE] If a Receiving Endpoint claims to support TS without TTS then it shall support both RTP payload format definitions MP2T and vnd.dlna.mpeg-mp2t.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	n/a	6T3VG	

NOTE: For full TS its up to the serving endpoint whether MP2T or vnd.dlna.mpeg-mp2t is used, this implies the Receiving Endpoint must be able to accept either one.

7.5.4.4.5.4.6 MT RTP clock accuracy for MPEG-2 TS and MPEG-2 PS

[GUIDELINE] The RTP timestamps applied to MPEG-2 TS and PS encapsulated content shall utilize a 90 KHz clock source that is accurate to within ± 30 parts per million (ppm) and have a slew rate of less than 0.075 Hz per second as defined in 13818-1 [5].

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[5] [25]	Y6YTG	

NOTE: Only that way the clock recovered by the Receiving Endpoint can meet the specifications needed for decoding.

7.5.4.4.5.4.7 MT RTP Target Transmission Time for MPEG TS and PS encapsulated content

[GUIDELINE] For payload types MP2P, MP2T, vnd.dlna.mpeg-mp2t, and vnd.dlna.mpeg-tts, the "Target Transmission Time" of an RTP packet is defined by its RTP Timestamp field.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	LCW9F	

7.5.4.4.5.5 Guidelines for encapsulation of WMA and WMV elementary streams

7.5.4.4.5.5.1 MT RTP ADU usage

[GUIDELINE] An ADU (Application Data Unit) shall be one complete MAU (Media Access Unit) as specified in [92].

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[92]	84D89	

7.5.4.4.5.5.2 MT RTP WMA time stamps in ASF are presentation time stamps

7.5.4.4.5.5.2.1

[GUIDELINE] The "Timestamp" field in the RTP header shall be set to a value reflecting the presentation time of the first payload in the RTP packet. The "Presentation Time" field of a WMA Media Object in ASF specifies the presentation time of the WMA Media Object.

[ATTRIBUTES]

M	L	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[91] [92]	T3VG6	C

7.5.4.4.5.5.2.2

[GUIDELINE] The "Decode Time" field shall not be included in the RTP Payload Format header, unless the decode time of the WMA media object is known.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[91], [92]	6YTGX	

NOTE: When WMA is stored in an ASF container, decode times of WMA Media Objects are not available.

7.5.4.4.5.5.3 MT RTP WMV time stamps in ASF are decode time stamps

7.5.4.4.5.5.3.1

[GUIDELINE] The "Timestamp" field in the RTP header shall be set to a value reflecting the decode time of the first payload in the RTP packet The "Presentation Time" field of a WMV Media Object in ASF specifies the decode timeof the WMV Media Object.

[ATTRIBUTES]

M	L	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[91] [92]	4D898	C

7.5.4.4.5.5.3.2

[GUIDELINE] The a=fmtp line in SDP shall specify ts=DTS to indicate this usage.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[92]	D898O	

7.5.4.4.5.5.3.3

[GUIDELINE] The "Presentation Time" field of a WMV media object in ASF shall not be used as the Presentation Time field of the RTP packet.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[91] [92]	3VG6O	

NOTE: When WMV is stored in an ASF container, presentation times of WMV Media Objects are not explicitly available.

7.5.4.4.5.5.4 MT RTP Determining the value of the SDP "profile" parameter for WMV

[GUIDELINE] If the ASF Meta Data Object exists, and if it contains a "DeviceConformanceTemplate" attribute for the WMV media stream, then that attribute may be used to determine the value for the "profile" parameter in the WMV SDP syntax.

The first two characters in the "DeviceConformanceTemplate" value determine the SDP "profile" parameter as follows:

- SP: profile=0
- MP:profile=1

[ATTRIBUTES]

O	C	DMS +PU+	M-DMS	n/a	[91] [92]	QS3WN	

7.5.4.4.5.5.5 MT RTP Determining the value of the SDP "level" parameter for WMV

[GUIDELINE] If the ASF Meta Data Object exists, and if it contains a "DeviceConformanceTemplate" attribute for the WMV media stream, then that attribute may be used to determine the value for the "level" parameter in the WMV SDP syntax.

The last two characters in the "DeviceConformanceTemplate" value determine the SDP "level" parameter as follows:

- LL: level=0
- ML: level=1
- HL: level=2

[ATTRIBUTES]

O	C	DMS +PU+	M-DMS	n/a	[91] [92]	ALCW9	

7.5.4.4.5.5.6 MT RTP Determining the value of the SDP "aspect" parameter for WMV

[GUIDELINE] If the ASF Meta Data Object exists, and if it contains both an "AspectRatioX" attribute and an "AspectRatioY" attribute for the WMV media stream, then those attributes can be used to determine the value for the "aspect" parameter in the WMV SDP syntax.

[ATTRIBUTES]

O	C	DMS +PU+	M-DMS	n/a	[91] [92]	PY6YT	

7.5.4.4.5.5.7 MT RTP Determining the value of the SDP "profile" parameter for WMA

[GUIDELINE] If the ASF Meta Data Object exists, and if it contains a "DeviceConformanceTemplate" attribute for the WMA media stream, then that attribute may be used to determine the value for the "profile" parameter in the WMA SDP syntax.

If the first character in the "DeviceConformanceTemplate" value is "L", then the second character may be used as the value for the SDP "profile" parameter.

[ATTRIBUTES]

O	C	DMS +PU+	M-DMS	n/a	[91] [92]	K6T3V	

7.5.4.4.5.6 Guidelines for the encapsulation for AVC elementary streams

7.5.4.4.5.6.1 MT RTP H.264 RTP Payload format

[GUIDELINE] If H.264/AVC elementary streams are transmitted over RTP, then RFC 3984 shall be used as specified in clauses 7.5.4.4.5.6.2 to 7.5.4.4.5.6.8 inclusive.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[43]	6QS3W	

7.5.4.4.5.6.2 MT RTP H.264 ADU usage

[GUIDELINE] An ADU shall be a complete a NAL unit.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[43]	ZALCW

7.5.4.4.5.6.3 MT RTP H.264 packetization

[GUIDELINE] A Serving Endpoint shall use one of the following packetization modes:

- single NAL unit mode,
- non-interleaved mode
- interleaved mode.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[43]	YPY6Y

NOTE: For the single NAL unit mode, non-interleaved mode, and interleaved mode, the values of the packetization-mode MIME/SDP parameter are equal to 0, 1, and 2, respectively.

7.5.4.4.5.6.4 MT RTP H.264 interleaved mode

7.5.4.4.5.6.4.1

[GUIDELINE] If rtp-h264-deint-buf-cap.dlna.org header (see guideline 7.5.4.4.5.6.7) is not present in the RTSP DESCRIBE request, then a Serving Endpoint shall set the value of sprop-interleaving-depth parameter in the a=fmtp line of the SDP equal to 0.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[43]	9ZALC

NOTE: When sprop-interleaving-depth equals to 0, the interleaved packetization mode can be used to encapsulate coded data from multiple coded pictures into the same RTP payload, without incurring any implementation complexity associated with interleaving of data. For low bit-rate media streams, this aggregation mechanism can help in avoiding a drop in the available bit rate for the whole WLAN segment and decreases the RTP/UDP/IP header overhead for the RTP stream.

7.5.4.4.5.6.4.2

[GUIDELINE] A Receiving Endpoint shall support the interleaved packetization mode with the value of the sprop-interleaving-depth parameter in the a=fmtp line of the SDP equal to 0.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[43]	H26QS

7.5.4.4.5.6.5 MT RTP H.264 single NAL unit mode

[GUIDELINE] A Receiving Endpoint shall support the single NAL unit packetization mode.

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[43]	26QS3

7.5.4.4.5.6.6 MT RTP H.264 non-interleaved mode

[GUIDELINE] A Receiving Endpoint shall support the non-interleaved packetization mode.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[43]	BWT85

7.5.4.4.5.6.7 MT RTP H.264 de-interleaving buffer capability

[GUIDELINE] If a Receiving Endpoint supports the interleaved packetization mode and a value of the sprop-interleaving-depth MIME/SDP parameter greater than 0, then the Receiving Endpoint shall include the rtp-h264-deint-buf-cap.dlna.org header in the DESCRIBE request. The syntax of the header is specified as follows:

- rtp-h264-deint-buf-cap.dlna.org = "rtp-h264-deint-buf-cap.dlna.org" *LWS ":" *LWS 1*DIGIT

The value of rtp-h264-deint-buf-cap.dlna.org shall be in the range of 0 to 4294967295, inclusive.

The value of the rtp-h264-deint-buf-cap.dlna.org header is interpreted identically to the deint-buf-cap MIME/SDP parameter specified in RFC 3984.

Example:

- rtp-h264-deint-buf-cap.dlna.org: 32768

This specifies that the Receiving Endpoint is capable of supporting such RTP streams that require a de-interleaving buffer up to 32768 bytes (inclusive).

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[43]	ZVEJZ	

NOTE: This indicates how big a buffer a Receiving Endpoint can allocate for the de-interleaving process of interleaved H.264 RTP streams.

7.5.4.4.5.6.8 MT RTP H.264 de-interleaving buffer requirement

[GUIDELINE] A serving endpoint shall set the value of the sprop-deint-buf-req parameter of H.264/AVC in the a=fmtp line of the SDP equal to or less than the value of the rtp-h264-deint-buf-cap.dlna.org header in the RTSP DESCRIBE request.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[43]	9RX99	

7.5.4.4.5.7 Guidelines for the encapsulation for MPEG-4 part 2 elementary streams

7.5.4.4.5.7.1 MT RTP payload format for MPEG-4 Part 2 elementary streams

[GUIDELINE] If MPEG-4 Part 2 elementary streams are transmitted over RTP, then the RTP payload format mpeg4-generic shall be used as specified in guidelines 7.5.4.4.5.7.2 to 7.5.4.4.5.7.5, inclusive.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[42]	G3BWT	

NOTE: "mpeg4-generic" is defined in RFC-3640.

7.5.4.4.5.7.2 MT RTP MPEG-4 Part 2 uses generic mode

[GUIDELINE] A Serving Endpoint that transmits MPEG-4 Part 2 elementary streams over RTP shall only use the "generic" mode of the RTP payload format mpeg4-generic.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[42]	RX99P	

NOTE: Generic mode is the only mode defined in RFC-3640 that is applicable to MPEG-4 Part 2.

The usage of generic mode is signaled by "mode=generic" on the a=fmtp line in SDP. See RFC 3640, subclause 3.3.2.

7.5.4.4.5.7.3 MT RTP MPEG-4 Part 2 ADU usage

[GUIDELINE] An ADU (Application Data Unit) shall be a complete AU (Access Unit) as specified in RFC 3640.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[42]	3BWT8	

7.5.4.4.5.7.4 MT RTP MPEG-4 Part 2 concatenation and fragmentation of Access Units

[GUIDELINE] A Serving Endpoint may concatenate and fragment MPEG-4 Access Units as defined in subclauses 2.3 and 2.4 of RFC-3640 [42].

[ATTRIBUTES]

O	C	DMS +PU+	M-DMS	n/a	[42]	W9NQT	

7.5.4.4.5.7.5 MT RTP MPEG-4 Part 2 interleaving of Access Units

[GUIDELINE] A Serving Endpoint shall not interleave MPEG-4 Access Units as defined in subclause 2.5 of RFC-3640. This means that if the "AU-Index-delta" field is present in the AU Header, then its value shall be set to 0. Additionally, if the SDP parameter maxDisplacement is present on the a=fmtp line, then its value shall be 0.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[42]	Q6Y3Q	

NOTE: Setting "AU-Index-delta" to 0 means that the index number of the Access Unit is equal to the previous index number plus one.

7.5.4.4.5.8 Guidelines for the encapsulation of MPEG-4 AAC streams

7.5.4.4.5.8.1 MT RTP payload format for MPEG-4 AAC and HE-ACC streams

[GUIDELINE] If MPEG-4 AAC or HE-ACC streams are transmitted over RTP, then the RTP payload format RFC 3640 shall be used as specified in guidelines 7.5.4.4.5.8.2 to 7.5.4.4.5.8.10 inclusive.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[42]	6Y3QA	C

NOTE: This guideline applies irrespective of the encoding tools used in the bit stream, such as AAC LC, LTP, BSAC, etc.

7.5.4.4.5.8.2 MT RTP MPEG-4 AAC

[GUIDELINE] A Serving Endpoint that transmits MPEG-4 AAC streams over RTP shall use the High Bit-rate AAC mode of the RTP payload format RFC 3640.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[42]	YZVEJ

NOTE: The usage of High Bit-rate AAC mode is signaled by "mode=AAC-hbr" on the a=fmtp line in SDP. See RFC 3640, subclause 3.3.5.

Alternative, low Bit-rate AAC mode could have been used only when AAC frame length is at most 63 octets, i.e., corresponding such a low bit rates that are used rarely in DLNA. Therefore low bit-rate mode is not allowed.

7.5.4.4.5.8.3 MT RTP MPEG-4 AAC concatenation and fragmentation of Access Units

7.5.4.4.5.8.3.1

[GUIDELINE] A Serving Endpoint may concatenate and fragment MPEG-4 AAC Access Units as defined in subclause 2.4 of RFC-3640.

[ATTRIBUTES]

O	R	DMS +PU+	M-DMS	n/a	[42]	T9RX9

7.5.4.4.5.8.3.2

[GUIDELINE] If a Serving Endpoint concatentates or fragments MPEG-4 AAC Access units, RTP packets shall not contain fragments of multiple Access Units.

[ATTRIBUTES]

M	R	DMS +PU+	M-DMS	n/a	[42]	5T9RX

7.5.4.4.5.8.4 MT RTP MPEG-4 AAC non-interleaving of Access Units

[GUIDELINE] A Serving Endpoint shall support non-interleaved packetization mode.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[42]	FYZVE

7.5.4.4.5.8.5 MT RTP MPEG-4 AAC interleaving of Access Units

[GUIDELINE] If a Serving Endpoint supports interleaving then it shall apply interleaving when rtp-aac-deint-buf-cap.dlna.org header is present in the RTSP DESCRIBE request.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[42]	XFYZV

7.5.4.4.5.8.6 MT RTP MPEG-4 AAC RTP interleaving in SDP

[GUIDELINE] A Serving Endpoint shall set the value of the maxDisplacement parameter in the a=fmtp line of the SDP equal to or less than the value of the rtp-aac-deint-buf-cap.dlna.org header in the RTSP DESCRIBE request.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[42]	XV7QD

NOTE: The value of the SDP maxDisplacement parameter in a=fmtp can be 0.

7.5.4.4.5.8.7 MT RTP MPEG-4 AAC non-interleaving of Access Units

[Guideline] When a serving endpoint uses non-interleaved mode then if the "AU-Index-delta" field is present in the AU Header, then its value shall be 0. Additionally, if the SDP parameter maxDisplacement is present on the a=fmtp line, then its value shall be 0.

[Attributes]

M	C	DMS +PU+	M-DMS	n/a	[42]	AW9NQ

NOTE: Setting "AU-Index-delta" to 0 means that the index number of the Access Unit is equal to the previous index number plus one.

7.5.4.4.5.8.8 MT RTP MPEG-4 AAC non-interleaved mode

[Guideline] A Receiving Endpoint shall support the non-interleaved packetization mode.

[Attributes]

M	A	DMP DMR	M-DMP	MIU	[42]	XQ6Y3

7.5.4.4.5.8.9 MT RTP MPEG-4 AAC interleaved mode

[Guideline] A Receiving Endpoint may support the interleaved packetization mode.

[Attributes]

O	A	DMP DMR	M-DMP	MIU	[42]	398XV

7.5.4.4.5.8.10 MT RTP MPEG-4 AAC RTP interleaving indication

[Guideline] If a Receiving Endpoint supports interleaving then it shall include the rtp-aac-deint-buf-cap.dlna.org header in the DESCRIBE request.

The syntax of this header is specified as follows:

- rtp-aac-deint-buf-cap-line = "rtp-aac-deint-buf-cap.dlna.org" *LWS ":" *LWS 1*DIGIT

The value of the rtp-aac-deint-buf-cap.dlna.org header is interpreted identically to the maxDisplacement SDP parameter, specified in [42].

[Attributes]

M	A	DMP DMR	M-DMP	MIU	[42]	QXQ6Y

7.5.4.4.5.9 Guidelines for the encapsulation of H.263 streams

7.5.4.4.5.9.1 MT RTP H.263 RTP Payload

[Guideline] If H.263 streams are transmitted over RTP, then the RTP payload format H263-2000 as defined in [40], and [30] shall be used with the constraints listed in guidelines 7.5.4.4.5.9.2 through 7.5.4.4.5.9.4, below.

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	[40] [30]	8XV7Q

7.5.4.4.5.9.2 MT RTP H.263 profile and level

[Guideline] A Serving Endpoint shall specify the profile and level in the a=fmtp line of SDP.

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	[30]	98XV7	

NOTE: The H.263 profile and level included in the a=fmtp line must follow the H263-2000 MIME media type specified in RFC 3555.

7.5.4.4.5.9.3 MT RTP H.263 frame size attribute at SDP media level

7.5.4.4.5.9.3.1

[Guideline] A Serving Endpoint shall specify the a=framesize attribute at the SDP media level for each H.263 stream in SDP.

The syntax is defined as:

- a-framesize = "a=framesize:" payload-type SP width "-" height
- payload-type = 1*DIGIT ; (shall be between 0 and 127)
- width = 1*DIGIT
- height = 1*DIGIT

Example:

- a=framesize:96 176-144

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	n/a	5XAW9	

7.5.4.4.5.9.3.2

[Guideline] Receiving Endpoints may support the a=framesize SDP attribute,

[Attributes]

O	A	DMP DMR	M-DMP	MIU	n/a	XAW9N	

7.5.4.4.5.9.4 MT RTP H.263 ADU usage

[Guideline] An ADU (Application Data Unit) shall be a complete video slice. Each RTP packet shall carry a single ADU.

[Attributes]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[40]	XG3BW	

7.5.4.4.5.10 Guidelines for the encapsulation AMR streams

7.5.4.4.5.10.1 MT RTP AMR RTP payload

[Guideline] If AMR streams are transmitted over RTP, then the RTP payload format RFC 3267 shall be used as specified in guidelines 7.5.4.4.5.10.2 to 7.5.4.4.5.10.4 inclusive.

[Attributes]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[36]	V7QD3	

7.5.4.4.5.10.2 MT RTP AMR RTP encapsulation

7.5.4.4.5.10.2.1

[GUIDELINE] A Serving Endpoint shall support encapsulation of one or more AMR speech frames into a single RTP packet and shall include maxptime attribute in SDP.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[36]	7QD33	

NOTE: The subclause 8.1 of [36] explains the SDP "maxptime" attribute.

7.5.4.4.5.10.2.2

[GUIDELINE] The recommended value for the SDP maxptime attribute is 200 msec.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	[36]	D33EI	

7.5.4.4.5.10.3 MT RTP AMR RTP interleaving

7.5.4.4.5.10.3.1

[GUIDELINE] A Serving Endpoint may support interleaving.

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	[36]	QD33E	

7.5.4.4.5.10.3.2

[GUIDELINE] If a Serving Endpoint supports interleaving then it shall apply interleaving when the rtp-amr-deint-buf-cap.dlna.org header is present in the RTSP DESCRIBE request.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[36]	VEJZD	

7.5.4.4.5.10.3.3

[GUIDELINE] A Serving Endpoint shall set the value of the interleaving parameter in the a=fmtp line of the SDP equal to or less than the value of the rtp-amr-deint-buf-cap.dlna.org header in the RTSP DESCRIBE request.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[36]	TIMZ3	

NOTE: The subclause 8.1 of [36] explains the SDP interleaving parameter.

The value of the SDP interleaving parameter in a=fmtp can be 0.

7.5.4.4.5.10.3.4

[GUIDELINE] A Receiving Endpoint may support interleaving.

[ATTRIBUTES]

O	A	DMP DMR	M-DMP	MIU	[36]	QARG4	

7.5.4.4.5.10.3.5

[GUIDELINE] If a Receiving Endpoint supports interleaving then it shall include the rtp-amr-deint-buf-cap.dlna.org header in the DESCRIBE request.

The syntax of this header is specified as follows:

- rtp-amr-deint-buf-cap-line = "rtp-amr-deint-buf-cap.dlna.org" *LWS ":" *LWS 1*DIGIT

The value of the rtp-amr-deint-buf-cap.dlna.org header is interpreted identically to the "interleaving" SDP parameter, specified in [36].

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[36]	EJZDY	

NOTE: The subclause 8.1 of [36] explains the SDP "interleaving" parameter.

7.5.4.4.5.10.4 MT RTP AMR RTP decapsulation

7.5.4.4.5.10.4.1

[GUIDELINE] A Receiving Endpoint shall be able to decapsulate an RTP packet having one or more AMR speech frames.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[36]	X99PE	

7.5.4.4.5.10.4.2

[GUIDELINE] A Receiving Endpoint should be able to support values of the SDP maxptime attribute of at least 200 msec.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[36]	Y3QAR	

NOTE: The subclause 8.1 of [36] explains maxptime attribute.

7.5.4.4.5.11 Guidelines for the encapsulation AMR-WBplus streams

7.5.4.4.5.11.1 MT RTP Payload format for AMR-WBplus streams

[GUIDELINE] If AMR-WBplus streams are transmitted over RTP, then the RTP payload format as specified in [31] shall be used as specified in guidelines 7.5.4.4.5.11.2 to 7.5.4.4.5.11.6 inclusive.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[31]	QTIMZ	

7.5.4.4.5.11.2 MT RTP AMR-WBplus encapsulation

7.5.4.4.5.11.2.1

[GUIDELINE] A Serving Endpoint shall support encapsulation of one or several AMR-WBplus frames into a single RTP packet and shall include maxptime attribute in SDP.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[31]	3QARG

7.5.4.4.5.11.2.2

[GUIDELINE] The recommended value for the SDP maxptime attribute is 200 msec.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	[31]	9NQTI

7.5.4.4.5.11.3 MT RTP AMR-WBplus basic mode

7.5.4.4.5.11.3.1

[GUIDELINE] A Serving Endpoint shall support basic mode.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[31]	NQTIM

7.5.4.4.5.11.3.2

[GUIDELINE] A Receiving Endpoint shall support basic mode.

[ATTRIBUTES]

M	R	DMP DMR	M-DMP	MIU	[31]	ARG45

7.5.4.4.5.11.4 MT RTP AMR-WBplus interleaving mode

7.5.4.4.5.11.4.1

[GUIDELINE] A Serving Endpoint may support interleaving mode.

[ATTRIBUTES]

O	L	DMS +PU+	M-DMS	n/a	[31]	ZDYQA

NOTE: This is limitation to [31] that mandates to support both basic and interleaving modes.

7.5.4.4.5.11.4.2

[GUIDELINE] If a Serving Endpoint supports interleaving mode then it shall use interleaving when the rtp-amrwbplus-deint-buf-cap.dlna.org header is present in the RTSP DESCRIBE request.

[ATTRIBUTES]

M	R	DMS +PU+	M-DMS	n/a	[31]	9PESQ

7.5.4.4.5.11.4.3

[GUIDELINE] A Serving Endpoint shall set the value of the "interleaving" parameter in the a=fmtp line of the SDP equal to or less than the value of the rtp-amrwbplus-deint-buf-cap.dlna.org header in the RTSP DESCRIBE request.

[ATTRIBUTES]

M	R	DMS +PU+	M-DMS	n/a	[31]	JZDYQ

NOTE: The subclauses 4.4 and 7.2.1 of [31] explains usage of SDP parameters for interleaving.

7.5.4.4.5.11.4.4

[GUIDELINE] If a Receiving Endpoint supports interleaving mode then it shall include rtp-amrwbplus-deint-buf-cap.dlna.org header in the DESCRIBE request.

The syntax of this header is specified as follows:

- rtp-amrwbplus-deint-buf-cap-line = "rtp-amrwbplus-deint-buf-cap.dlna.org" *LWS ":" *LWS 1*DIGIT

The value of this header is interpreted identically to the "interleaving" SDP parameter, specified in [31].

[ATTRIBUTES]

M	R	DMP DMR	M-DMP	MIU	[31]	99PES	

NOTE: The value of interleaving parameter must be greater than zero (see 7.2.1 of [31]). If it is not present, interleaving mode must not be used,

The subclauses 4.4 and 7.2.1 of [31] explains usage of SDP parameters for interleaving.

7.5.4.4.5.11.5 MT RTP AMR-WBplus RTP decapsulation

7.5.4.4.5.11.5.1

[GUIDELINE] A Receiving Endpoint shall be able to decapsulate an RTP packet having one or more AMR-WBplus frames.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[31]	8592P	

7.5.4.4.5.11.5.2

[GUIDELINE] A Receiving Endpoint should be able to support values of the SDP maxptime attribute of at least 200 msec.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[31]	3WNRR	

7.5.4.4.5.11.6 MT RTP AMR-WBplus channels

[GUIDELINE] A Receiving Endpoint that is only capable of playout of monophonic audio shall still accept signals originally encoded and transmitted as stereo.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[31]	CW9FJ	

NOTE: The AMR-WBplus decoder has the capability of stereo to mono downmixing as part of the decoding process.

7.5.4.4.6 RTP Media Transport: RTSP for control of RTP streams

7.5.4.4.6.1 General

The Real Time Streaming Protocol (RTSP) provides a standard protocol for controlling media streams, including starting, stopping and pausing. Media operations such as fast forward scan, slow backward scan, etc can also be implemented in a standard way using RTSP. RTSP is the required protocol for controlling streams for RTP transport.

The Session Description Protocol (SDP) provides a standard method of communicating RTP session parameters. It is used as the format of the description of the media that is sent from the Serving Endpoint to the Receiving Endpoint in the response to the RTSP DESCRIBE method.

The DLNA RTSP and SDP signaling in 7.5.4.4.6.2 RTP Media Transport: RTSP/SDP Protocols specifies a minimal strict subset of RTSP (RFC 2326 [27]) and SDP (RFC 2327 [28]) to be implemented by DLNA Serving Endpoints and Receiving endpoints.

7.5.4.4.6.2 RTP Media Transport: RTSP/SDP Protocols

7.5.4.4.6.2.1 MT RTP RTSP support

[GUIDELINE] A DLNA device shall implement RTSP version 1.0 as the mandatory media transport control protocol as described in Appendix D of RFC 2326 with constraints and extensions defined in subsequent entries of this table. Specifically serving endpoints shall implement RTSP servers and Receiving Endpoints shall implement RTSP clients.

[ATTRIBUTES]

M	L	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[27]	PESQI	

NOTE: RTSP controls the RTP media transport.

7.5.4.4.6.2.2 MT RTP RTSP SDP support

[GUIDELINE] A DLNA device shall implement SDP as the mandatory format used in the RTSP DESCRIBE method.

[ATTRIBUTES]

M	R	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[28]	T8592	

NOTE: SDP declares the properties of the media streams that can be present in the RTSP session.

7.5.4.4.6.2.3 MT RTP RTSP TCP Transport

7.5.4.4.6.2.3.1

[GUIDELINE] Both Serving Endpoint and Receiving Endpoint devices shall use TCP to transport RTSP.

[ATTRIBUTES]

M	L	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[27]	WT859	

NOTE: RTSP/UDP usage is not allowed.

7.5.4.4.6.2.3.2

[GUIDELINE] A Serving Endpoint should support persistent TCP connections. This means that the TCP connection should not be closed by Serving Endpoint until after all RTSP sessions using the TCP connection have been terminated.

[ATTRIBUTES]

S	C	DMS +PU+	M-DMS	n/a	[27]	S3WNR	

7.5.4.4.6.2.3.3

[Guideline] A Receiving Endpoint should use a persistent TCP connection for each RTSP session. This means that the TCP connection should not be closed by Receiving Endpoint until after the RTSP session has been terminated.

[Attributes]

S	C	DMP DMR	M-DMP	MIU	[27]	592PB	

7.5.4.4.6.2.4 MT RTP RTSP pipelined requests

[Guideline] Serving Endpoint shall accept pipelined RTSP requests, and shall respond to RTSP requests in the order that they were received.

[Attributes]

M	C	DMS +PU+	M-DMS	n/a	[27]	NRRAN	

7.5.4.4.6.2.5 MT RTP Multiple RTSP sessions

[Guideline] Serving endpoints should support more than one simultaneous RTSP session.

[Attributes]

S	C	DMS +PU+	M-DMS	n/a	[27]	9FJ7Y	

NOTE: It is a good practice for a serving endpoint to support multiple Receiving Endpoints simultaneously.

7.5.4.4.6.2.6 MT RTP RTSP session multiplexing

[Guideline] Receiving Endpoint shall use a separate TCP connection for each RTSP session.

[Attributes]

M	L	DMP DMR	M-DMP	MIU	[27]	GXOS9	

7.5.4.4.6.2.7 MT RTP Receive RTSP requests from a Serving Endpoint

7.5.4.4.6.2.7.1

[Guideline] A Receiving Endpoint shall support receiving RTSP requests from the Serving Endpoint over the same TCP connection that the Receiving Endpoint is using for sending requests to the Serving Endpoint.

[Attributes]

M	C	DMP DMR	M-DMP	MIU	[27]	WNRRA	

7.5.4.4.6.2.7.2

[Guideline] If a Receiving Endpoint does not implement support for an RTSP request type, it shall respond to a Serving Endpoint with a "501 Not Implemented" status code.

[Attributes]

M	C	DMP DMR	M-DMP	MIU	[27]	TGXOS	

7.5.4.4.6.2.8 MT RTP Indication of supported RTSP version

7.5.4.4.6.2.8.1

[GUIDELINE] Serving Endpoints and Receiving Endpoints should specify the highest RTSP version number that they are compliant with when sending a RTSP response regardless of which RTSP version number was specified in the corresponding RTSP request.

[ATTRIBUTES]

S	C	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[27]	G6O7W

7.5.4.4.6.2.8.2

[GUIDELINE] If a Serving Endpoint or a Receiving Endpoint receives a RTSP request that specifies a higher version of RTSP than is supported, then it shall respond with status code 505 ("RTSP version not supported").

[ATTRIBUTES]

M	C	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[27]	W9FJ7

7.5.4.4.6.2.8.3

[GUIDELINE] Serving Endpoints and Receiving Endpoints shall accept RTSP responses that specify a RTSP version number higher than the highest supported RTSP version.

[ATTRIBUTES]

M	C	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[27]	YTGXO

7.5.4.4.6.2.9 MT RTP Tolerate unknown RTSP headers and SDP attributes

[GUIDELINE] Receiving Endpoints and Serving Endpoints shall be tolerant of unknown RTSP headers and SDP attributes.

Tolerant behavior is defined as being able to successfully "parse and interpret" or "parse and ignore" the RTSP headers, their values and SDP attributes and their values.

[ATTRIBUTES]

M	R	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[27] [28]	6O7W7

NOTE: This guideline addresses forward compatibility and allows for broader interoperability with implementations that employ transport layer vendor extensions by way of RTSP headers.

7.5.4.4.6.2.10 MT RTP RTSP Case-sensitivity

[GUIDELINE] Names of RTSP methods shall be treated as case sensitive tokens, while the RTSP headers shall be treated as case-insensitive tokens.

[ATTRIBUTES]

M	C	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[27]	VG6O7

NOTE: This is normative according to the RTSP specification.

7.5.4.4.6.2.11 MT RTP RTSP Required requests

7.5.4.4.6.2.11.1

[GUIDELINE] A Serving Endpoint shall support receiving the following RTSP requests:

- DESCRIBE
- SETUP
- PLAY
- OPTIONS
- TEARDOWN

A Serving Endpoint need not support receiving any other RTSP requests.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[27]	FJ7YD	

7.5.4.4.6.2.11.2

[GUIDELINE] A Receiving Endpoint shall not depend on the support of any RTSP request not listed in guideline 7.5.4.4.6.2.11.1.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[27]	XOS9X	

7.5.4.4.6.2.11.3

[GUIDELINE] A Receiving Endpoint shall support sending the following RTSP requests:

- DESCRIBE
- SETUP
- PLAY
- OPTIONS
- TEARDOWN

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[27]	98OYB	

7.5.4.4.6.2.12 MT RTP DLNA Media Operations Summary

[GUIDELINE] The DLNA Streaming Media Operations (in Table 46) shall be implemented as defined in the following guidelines:

- Play: 7.5.4.4.6.2.36 MT RTP Play Media Operation
- Stop: 7.5.4.4.6.2.55 MT RTP Stop Media Operation
- Pause: 7.5.4.4.6.2.47 MT RTP Pause Media Operation
- Pause-Release: 7.5.4.4.6.2.49 MT RTP Pause-Release Media Operation
- Seek: 7.5.4.4.6.2.39 MT RTP Seek Media Operation
- Fast Forward Scan7.5.4.4.6.2.43 MT RTP Scan Media Operations
- Slow Forward Scan: 7.5.4.4.6.2.43 MT RTP Scan Media Operations
- Fast Backward Scan: 7.5.4.4.6.2.43 MT RTP Scan Media Operations

- Slow Backward Scan: 7.5.4.4.6.2.43 MT RTP Scan Media Operations
- Streaming Download: n/a

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	n/a	O7W77	

7.5.4.4.6.2.13 MT RTP CSeq header

[GUIDELINE] A Serving Endpoint shall include the CSeq header in all RTSP responses, and in any RTSP requests that it sends.

The CSeq header in requests sent by the Serving Endpoint shall be independent of the CSeq header in requests sent by the Receiving Endpoint.

[ATTRIBUTES]

M	R	DMS +PU+	M-DMS	n/a	[27]	898OY	

7.5.4.4.6.2.14 MT RTP Supported header

[GUIDELINE] The Supported header field shall comply with the following syntax::

- supported-line = "Supported" *LWS ":" *LWS [feature-tag *(*LWS "," *LWS feature-tag)]
- feature-tag = token

Examples:

- Supported: dlna.announce
- Supported: dlna.announce, rtsp.basic

The RTSP header name is the "Supported" string and the header value is zero or more comma-separated tokens. Both header name and value are treated as case insensitive strings.

Note that the Supported header can be used by Serving Endpoints and Receiving Endpoints issuing an RTSP request on non-DLNA content.

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[35]	VJ9KN	

NOTE: The Supported header is currently defined for the SIP protocol (see [35], subclause 20.37.)

7.5.4.4.6.2.15 MT RTP Event-Type.dlna.org header

[GUIDELINE] The Event-Type.dlna.org header shall comply with the following syntax:

- event-type-line = "Event-Type.dlna.org" *LWS ":" *LWS event code
- event-code = 4DIGIT ; 0-9999

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	n/a	J9KNJ	

- in

NOTE: This header is used with the ANNOUCE request (guideline 7.5.4.4.6.2.52.1) to indicate an event that has occurred at the Serving Endpoint.

The event code consists of exactly 4 numerical digits.

7.5.4.4.6.2.16 MT RTP Available-Range.dlna.org header

7.5.4.4.6.2.16.1

[Guideline] The Available-Range.dlna.org header shall comply with the following syntax:

- available-range-line = "Available-Range.dlna.org" *LWS ":" *LWS "npt" "=" npt-time "-" npt-time
- npt-time = <syntax as defined in 7.5.4.2.13>

[Attributes]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[27]	8OYB8

7.5.4.4.6.2.16.2

[Guideline] The first npt-time parameter should correspond to the smallest NPT time value that Receiving Endpoint may specify as the start time in a Range header.

The second npt-time parameter should correspond to the largest NPT time value that Receiving Endpoint may specify as the start time in a Range header.

[Attributes]

S	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	n/a	MVJ9K

7.5.4.4.6.2.17 MT RTP Buffer-Info.dlna.org header

[Guideline] The Buffer-Info.dlna.org header shall comply with the following syntax:

- buffer-info-line = "Buffer-Info.dlna.org" *LWS ":" *LWS "dejitter" "=" 1*10DIGIT [";" cdb-params] [";" td-params] [";"bfr-params]
- cdb-params = "CDB" "=" 1*10DIGIT ";" "BTM" "=" "0" | "1" | "2"
- td-params = "TD" "=" 1*10DIGIT
- bfr-params = "BFR" "=" "0" | "1"

Note that the literals, "dejitter", "CDB", "BTM", "TD", and "BFR", are case sensitive.

The first parameter, "dejitter", specifies the size, in bytes, of the network de-jitter buffer.

If cdb-params is present, the "CDB" parameter specifies the size of the Coded Data Buffer in bytes. The BTM parameter specifies the buffer transfer mechanism between the network de-jitter buffer and the coded data buffer. The BTM parameter shall comply with the following syntax:

- 0: When the coded data buffer has empty space then the de-jitter buffer will transfer the data immediately.
- 1: The data is transferred according to packet's timestamp.
- 2: Other transfer mechanism than the above.

If td-params is present, the TD (Target Buffer Duration) parameter specifies the minimal amount of data that is required for interrupt free playback in the combination of the Receiving Endpoint's network de-jitter buffer and the Coded Data Buffer (if any). The value of the TD parameter is represented in millisecond time units.

If bfr-params is present, the BFR parameter specifies if the Receiving Endpoint will transmit Buffer Fullness Reports.The value 1 means that the Receiving Endpoint will transmit Buffer Fullness Reports and the value 0 indicates that the Receiving Endpoint will not transmit Buffer Fullness Reports.

Example:

- Buffer-Info.dlna.org: dejitter=65536;CDB=98302;BTM=0;TD=1000;BFR=1

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	n/a	9KNJD	

NOTE: The Coded Data Buffer (CDB) contains only raw compressed data bit stream without RTP headers.

If the CDB is present on the Receiving Endpoint and the buffer transfer mechanism is not signaled, the input to the coded data buffer shall follow an appropriate buffering model. In other words, when a Receiving Endpoint maintains a coded data buffer for an audio elementary stream, coded audio frames are moved to the CDB according to their RTP timestamp. When a Receiving Endpoint maintains a coded data buffer for a video elementary stream, the input to the CDB is done as specified in the Hypothetical Reference Decoder specification of the video coding profile in use. When a Receiving Endpoint maintains a coded data buffer for a MPEG-2 TS or MPEG-2 PS, RTP payloads are input to the CDB according to their RTP timestamps.

The Target Buffer Duration indicated by a Receiving Endpoint informs the Serving Endpoint to fill up the receiver buffer (network de-jitter buffer + Coded Data Buffer) to a minimum amount of data to prevent buffer underflow.

If the amount of data associated with the Target Buffer Duration is larger than the size of the receiver buffer then it does not imply that the Serving Endpoint is allowed to exceed size of the receiver buffer.

The Serving Endpoint must not intentionally overflow the Receiving Endpoint's receiver buffer at any time.

The Serving Endpoint might need to adjust the encoding rate to fulfill the Target Buffer Duration indication while maintaining the Receiving Endpoint's receiver buffer.

7.5.4.4.6.2.18 MT RTP WCT.dlna.org header

[GUIDELINE] The WCT.dlna.org header shall comply with the following syntax:

- wct-line = "WCT.dlna.org" *LWS ":" *LWS <val>
- <val> = "0" | "1'"

Example:

- WCT.dlna.org: 1

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	n/a	OYB8U	

NOTE: This header is used to request Serving Endpoint to add Wall Clock Time samples.

7.5.4.4.6.2.19 MT RTP Max-Prate.dlna.org header

[GUIDELINE] The Max-Prate.dlna.org RTSP header shall comply with the following syntax:

- max-prate-line = "Max-Prate.dlna.org" *LWS ":" *LWS max-packet-rate
- max-packet-rate = 1*DIGIT
- max-packet-rate is expressed in units of packets per second (pps).

Example:

- Max-Prate.dlna.org: 200

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	n/a	ULNQC	

NOTE: The Receiving Endpoint can include the header Require: dlna.Max-Prate in a request to determine if the Serving Endpoint takes the Max-Prate.dlna.org header into account. (See guideline 7.5.4.4.6.2.29.2.)

7.5.4.4.6.2.20 MT RTP RTSP Require header

[GUIDELINE] If the Serving Endpoint receives a request with a Require header and that header lists at least one unrecognized feature tag, then the Serving Endpoint shall respond with a status code 551 (Option Not Supported).

[ATTRIBUTES]

M	R	DMS +PU+	M-DMS	n/a	[27]	D24UR	

7.5.4.4.6.2.21 MT RTP RTSP aggregate control

7.5.4.4.6.2.21.1

[GUIDELINE] The Serving Endpoint shall support aggregate control. This implies that error code 459 shall not be returned by PAUSE, PLAY and TEARDOWN methods.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[27]	75Y4A	

NOTE: For RTSP sessions consisting of multiple RTP streams, the RTSP protocol supports most methods at both the presentation (= aggregate) and at the individual RTP stream (= non-aggregate) level.

However, PLAY, PAUSE and TEARDOWN is most appropriately done at the aggregate level.

7.5.4.4.6.2.21.2

[GUIDELINE] The Receiving Endpoint shall support aggregate control for PAUSE, PLAY, and TEARDOWN commands.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[27]	X3TVQ	

7.5.4.4.6.2.22 MT RTP Send RTCP reports when in RTSP Ready state

7.5.4.4.6.2.22.1

[GUIDELINE] If a Receiving Endpoint is sending RTCP reports, it shall do so while the RTSP session is in the Ready and Playing states if at least one RTP packet has been received.

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[27]	DHHKD	

NOTE: When the RTSP session is paused, RTSP enters Ready state. (The RTSP state machine is defined in RFC-2326.) RTCP reports must be sent even when the RTSP session is paused.

7.5.4.4.6.2.22.2

[GUIDELINE] If a Serving Endpoint is sending RTCP reports, it shall do so while the RTSP session is in the Ready and Playing states if at least one RTP packet has been transmitted.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[27]	775Y4	

NOTE: RTCP reports must be sent even when the RTSP session is paused.

If a Serving Endpoint is relaying an RTP stream, it might not know the SSRC of the RTP stream (and will be unable to send an RTCP Sender Report) until it has it has relayed the first RTP packet for that stream.

7.5.4.4.6.2.23 MT RTP Serving Endpoint Timeout (keep alive)

7.5.4.4.6.2.23.1

[GUIDELINE] A Serving Endpoint shall terminate the RTSP session if it has received neither an RTCP Receiver Report nor an RTSP request within a timeout interval.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[27]	NDS9M	

7.5.4.4.6.2.23.2

[GUIDELINE] The recommended timeout interval for receiving RTCP Receiver Reports or RTSP requests is 30 seconds.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	n/a	JD24U	

7.5.4.4.6.2.23.3

[GUIDELINE] For timeout interval values other than 60 seconds the Serving Endpoint shall use the "timeout" parameter in the RTSP Session header to communicate the value of the timeout interval to the Receiving Endpoint.

[ATTRIBUTES]

M	R	DMS +PU+	M-DMS	n/a	[27]	B8ULN	

NOTE: If the "timeout" parameter is not specified, the timeout interval defaults to 60 seconds.

7.5.4.4.6.2.23.4

[GUIDELINE] A Receiving Endpoint shall send at least one RTSP request with the Session header per timeout interval.

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[27]	NJD24	

NOTE: Commands such as SETUP, PLAY and PAUSE alter the RTSP session state, while OPTIONS does not.

The recommended "keep alive" method is to send a RTSP OPTIONS request with Session header. RTCP can also be used if implemented.

7.5.4.4.6.2.23.5

[GUIDELINE] A Receiving Endpoint should send an OPTIONS request with the Session header per timeout interval if no other RTSP request needs to be sent during a given timeout interval.

[ATTRIBUTES]

S	L	DMP DMR	M-DMP	MIU	[27]	8ULNQ	

7.5.4.4.6.2.24 MT RTP Receiving Endpoint timeout (keep alive)

7.5.4.4.6.2.24.1

[GUIDELINE] If a Receiving Endpoint has not received an RTP or RTCP packet from the Serving Endpoint within a timeout interval and the RTSP sess ion is in the Playing state, the Receiving Endpoint should terminate the RTSP session as described in guideline 7.5.4.4.6.2.55.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[27]	KNJD2	

NOTE: Some RTSP servers might not send RTCP packets while in Paused state, so this guideline applies to Playing state only.

When choosing a timeout interval, a Receiving Endpoint is exptected to consider the streaming bit rate and the bit rate used for RTCP reports. (Low bit rates might require a larger timeout interval.)

7.5.4.4.6.2.24.2

[GUIDELINE] A Receiving Endpoint should assume that the RTSP session has been terminated if it has not received an RTSP command response within a timeout interval.

The recommended value for this timeout interval is 30 seconds.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[27]	YB8UL	

7.5.4.4.6.2.24.3

[GUIDELINE] If the RTSP session is assumed to have been terminated, the Receiving Endpoint should close the RTSP TCP connection to the Serving Endpoint.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[27]	W775Y	

7.5.4.4.6.2.25 MT RTP Session description format used in DESCRIBE

7.5.4.4.6.2.25.1

[GUIDELINE] A Receiving Endpoint shall specify the MIME content type "application/sdp" in the Accept header of a DESCRIBE request.

[ATTRIBUTES]

M	L	DMP DMR	M-DMP	MIU	[27]	7W775	

NOTE: The Receiving Endpoint can request additional content types using the Accept header, but the Serving Endpoint might not support any additional content types.

7.5.4.4.6.2.25.2

[GUIDELINE] If the Serving Endpoint receives a DESCRIBE request and the request does not contain an Accept header, then the Serving Endpoint shall infer that "application/sdp" is the only MIME content type supported by the Receiving Endpoint.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[27]	ANDS9	

7.5.4.4.6.2.25.3

[GUIDELINE] If the Serving Endpoint responds to a DESCRIBE request with status code 200 ("OK"), then the response shall include the following:

- An entity body in one of the formats specified in the Accept header of the DESCRIBE request, or in SDP format if no Accept header was included.
- A Content-Type header, and this header shall specify the MIME content type of the entity body.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[27]	7YDHH	

NOTE: Since Receiving Endpoints always specify "application/sdp" as one of the MIME content types, this means that the Serving Endpoint must always support respdonding with an entity body in SDP format.

7.5.4.4.6.2.26 MT RTP Available-Range.dlna.org header in DESCRIBE response

7.5.4.4.6.2.26.1

[GUIDELINE] If a Serving Endpoint supports the Limited Random Access Data Availability model for a content binary (defined in 7.4.1.3.28), then it shall include the Available-Range.dlna.org header (guideline 7.5.4.4.6.2.16) in the DESCRIBE response.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	n/a	OS9X3	

7.5.4.4.6.2.26.2

[GUIDELINE] If the Available-Range.dlna.org header is included in the DESCRIBE response, the first npt-time parameter in that header should be set to the UCDAM's data position of of r_0, and the second npt-time parameter should be set to the UCDAM's data position of r_N.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	n/a	9X3TV	

NOTE: The variables r_0 and r_N, represent the earliest time and latest time, respectively, in the RTP stream (or media stream) that the Receiving Endpoint can seek to.

7.5.4.4.6.2.27 MT RTP/BCM RTSP peerManager.dlna.org

7.5.4.4.6.2.27.1

[GUIDELINE] A Receiving Endpoint should include the header peerManager.dlna.org in an RTSP DESCRIBE request to communicate the identity of the UPnP AV ConnectionMananger service to the Serving Endpoint.

[ATTRIBUTES]

S	A	DMR	n/a	n/a	[45]	S9X3T	

NOTE: The header peerManager.dlna.org is not required. However, this feature is recommended to help a Serving Endpoint that implements BCM.

7.5.4.4.6.2.27.2

[GUIDELINE] The value of the header peerManager.dlna.org shall be the same value as the PeerConnectionManager, as described in the UPnP AV Architecture.

The notation of the peerManager.dlna.org RTSP header is the same as peerManager-line, defined in 7.5.4.3.2.36.2.

[Attributes]

M	A	DMR	n/a	n/a	[45]	YDHHK

7.5.4.4.6.2.28 MT RTP/BCM RTSP friendlyName.dlna.org

7.5.4.4.6.2.28.1

[Guideline] A Receiving Endpoint should include the header friendlyName.dlna.org in an RTSP DESCRIBE request to communicate a user-friendly name to the Serving Endpoint.

[Attributes]

S	A	DMP	M-DMP	MIU	n/a	J7YDH

NOTE: The header friendlyName.dlna.org is not required. However, this feature is recommended to help a Serving Endpoint that implements BCM.

7.5.4.4.6.2.28.2

[Guideline] The notation of the friendlyName.dlna.org RTSP header is the same as the friendlyName-line, defined in 7.5.4.3.2.37.2 (MT/BCM HTTP Header:friendlyName.dlna.org).

[Attributes]

M	A	DMP	M-DMP	MIU	n/a	RANDS

7.5.4.4.6.2.29 MT RTP Maximum reception packet rates

7.5.4.4.6.2.29.1

[Guideline] A Receiving Endpoint may include the header Max-Prate.dlna.org (defined in guideline 7.5.4.4.6.2.19) in an RTSP DESCRIBE request.

The header Max-Prate.dlna.org declares the maximum packet data rate capability of the Receiving Endpoint.

The header may be included in any subsequent RTSP request, in case the Receiving Endpoint wants to update its maximum reception packet rate capability during a RTSP session.

[Attributes]

O	A	DMP DMR	M-DMP	MIU	n/a	A6ZWY

NOTE: This rule enables the signaling of the maximum Receiving Endpoint reception packet rate to inform the Serving Endpoint.

It avoids a lightweight Receiving Endpoint having performance problems or losing packets in case the incoming packet rate is too high to be sustained.

This guideline does not enable a Receiving Endpoint to change/override the supported and/or negotiated media profile(s) for a particular RTSP session.

7.5.4.4.6.2.29.2

[Guideline] If a Receiving Endpoint includes the Max-Prate.dlna.org in an RTSP request, then it should also include the Require: dlna.Max-Prate header in the same RTSP request.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[27]	QIR34	

NOTE: This allows the Receiving Endpoint to get an acknowledgment from the Serving Endpoint.

7.5.4.4.6.2.30 MT RTP Supported header when RTP packet retransmission is supported

7.5.4.4.6.2.30.1

[GUIDELINE] If a Receiving Endpoint supports retransmitted packets formatted using the RTP payload formats audio/rtx and video/rtx, as defined in [10] ,then Receiving Endpoint shall include the Supported header with feature-tag dlna.rtx in the DESCRIBE request.

Example:

- Supported: dlna.rtx

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[10]	2PBOY	

NOTE: The indication of which retransmission methods, if any, are supported, allows Serving Endpoint to decide if it will accept RTCP nack feedback messages, and allows it to indicate this decision in the SDP a=rtcp-fb parameter.

7.5.4.4.6.2.30.2

[GUIDELINE] If a Receiving Endpoint supports retransmitted packets sent as an identical copy of the original packet to the same UDP port as the original packet, then the Receiving Endpoint shall include the Supported header with feature-tag dlna.rtx-dup in the DESCRIBE request.

Example:

- Supported: dlna.rtx-dup

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	n/a	RRAND	

7.5.4.4.6.2.31 MT RTP SETUP request Clarifications

[GUIDELINE] While in Playing state, a Serving Endpoint shall support the SETUP method, to allow a Receiving Endpoint to change RTP and RTCP ports of the RTP Session.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[27]	BOYV9	

NOTE: The RTSP protocol state machine is described in RFC-2326, subclause A.1.

7.5.4.4.6.2.32 MT RTP Transport header when RTP/AVPF is used

[GUIDELINE] If the RTP/AVPF profile is used, the Receiving Endpoint and the Serving Endpoint shall specify the transport field in the Transport header as "RTP/AVPF".

[ATTRIBUTES]

M	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[8] [27]	92PBO	

7.5.4.4.6.2.33 MT RTP Tolerate RTP/AVP when RTP/AVPF is expected

7.5.4.4.6.2.33.1

[GUIDELINE] A Serving Endpoint shall accept Transport headers that specify the transport field as "RTP/AVP" when "RTP/AVPF" is expected.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[8] [27]	SQIR3	

NOTE: Receiving Endpoints that do not support RTP/AVPF can specify RTP/AVP in the Transport header.

7.5.4.4.6.2.33.2

[GUIDELINE] A Receiving Endpoint shall accept Transport headers that specify the transport field as "RTP/AVPF" when "RTP/AVP" is expected.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[8] [27]	PBOYV	

7.5.4.4.6.2.34 MT RTP Buffer headers in SETUP

7.5.4.4.6.2.34.1

[GUIDELINE] If the Serving Endpoint indicates that bfr is an acceptable RTCP feedback message type for an RTP stream, and the Receiving Endpoint will transmit Buffer Fullness Reports for that RTP stream, then it shall include the Buffer-Info.dlna.org header (guideline 7.5.4.4.6.2.17) in the RTSP SETUP request. Furthermore, the BFR parameter shall be present on the Buffer-Info.dlna.org header, and its value shall be set to 1.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	n/a	ESQIR	

NOTE: The Serving Endpoint can use the network de-jitter buffer size and the pre-decoder buffer size along with buffer fullness reported in the bfr RTCP feedback message to compute buffer levels at the RENEDERING ENDPOINT. See also guideline 7.5.4.4.3.13.

7.5.4.4.6.2.34.2

[GUIDELINE] A Receiving Endpoint may send the Buffer-Info.dlna.org header (guideline 7.5.4.4.6.2.17) in the RTSP SETUP request.

[ATTRIBUTES]

O	A	DMP DMR	M-DMP	MIU	n/a	IR346	

NOTE: Even if BFR's will not be used, the Receiving Endpoint can still include the Buffer-Info.dlna.org header in the SETUP request. This is particularly useful if the size of the pre-decoder buffer is smaller than specified in the a=predecbufsize.dlna.org SDP attribute, because it can allow Serving Endpoint to adapt the encoding rate to match the smaller pre-decoder buffer size.

7.5.4.4.6.2.35 MT RTP PLAY requests shall not be sent in Playing state

7.5.4.4.6.2.35.1

[GUIDELINE] A Receiving Endpoint shall send a PAUSE request between each PLAY request in an RTSP session.

[ATTRIBUTES]

M	L	DMP DMR	M-DMP	MIU	[27]	45P7Q

NOTE: The PAUSE request is required to bring the RTSP protocol state machine from the Playing state to the Ready state. (A PLAY request sent while in the Playing state would cause the serving endpoint to interpret it as a "queued PLAY" request, as defined in subclause 10.5 of RFC-2326.)

7.5.4.4.6.2.35.2

[GUIDELINE] If the response to a PAUSE request indicates error code 455 ("Method Invalid in the current state"), then this error should be ignored by the Receiving Endpoint.

[ATTRIBUTES]

S	L	DMP DMR	M-DMP	MIU	[27]	IAT8S

NOTE: Some Serving Endpoints can enter Ready state at the end of the RTP stream. In that case they might respond to a PAUSE request with a 455 error. This can be safely ignored by the Receiving Endpoint.

7.5.4.4.6.2.35.3

[GUIDELINE] If the response to a PAUSE request indicates error code 501 ("Not Implemented"), then the Receiving Endpoint shall send a TEARDOWN request to destroy the RTSP session, followed by creating a new RTSP session.

[ATTRIBUTES]

M	L	DMP DMR	M-DMP	MIU	[27]	Z3MS6

NOTE: If PAUSE is not supported by the Serving Endpoint, the only way to bring the RTSP protocol state machine to Ready state is by sending a TEARDOWN, followed by a new DESCRIBE and new SETUP requests.

7.5.4.4.6.2.36 MT RTP Play Media Operation

7.5.4.4.6.2.36.1

[GUIDELINE] If the RTSP protocol state machine of the Receiving Endpoint is in the Ready state, then it shall implement the Play Media Operation by sending an RTSP PLAY request.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[27]	QA6ZW

NOTE: The RTSP protocol state machine is described in RFC-2326, subclause A.1.

7.5.4.4.6.2.36.2

[GUIDELINE] If the RTSP protocol state machine of the Receiving Endpoint is not in the Ready state, then it shall implement the Play Media Operation by bringing the RTSP protocol state machine into the Ready state, followed by sending a PLAY request as described in guideline 7.5.4.4.6.2.36.1.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[27]	YQA6Z

NOTE: To bring the state machine into Ready state, it might be necessary to send one or more SETUP requests. In order to be able to send SETUP requests, the Receiving Endpoint might have to send a DESCRIBE request. If the state machine is in Playing state, guideline 7.5.4.4.6.2.36.1 describes how to reach Ready state.

7.5.4.4.6.2.37 MT RTP Rtp-Info header

[Guideline] If valid seq and rtptime parameters are available or can be determined, the Serving Endpoint shall include an Rtp-Info header with those parameters in the response to the PLAY request.

[Attributes]

M	L	DMS +PU+	M-DMS	n/a	[27]	DYQA6	

7.5.4.4.6.2.38 MT RTP Wall Clock Time sample request

7.5.4.4.6.2.38.1

[Guideline] If a Receiving Endpoint wants Wall Clock Time samples to be included in the RTP packets, then it shall include the header WCT.dlna.org: 1 (defined in guideline 7.5.4.4.6.2.18) in the RTSP PLAY request.

[Attributes]

M	A	DMP DMR	M-DMP	MIU	n/a	RG45P	

NOTE: Wall Clock Time samples are defined in guideline 7.5.4.4.3.15.

7.5.4.4.6.2.38.2

[Guideline] If a Serving Endpoint supports Wall Clock Time samples in accordance with guideline 7.5.4.4.3.15, then it shall understand the WCT.dlna.org header (defined in guideline 7.5.4.4.6.2.18) in a RTSP PLAY request.

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	n/a	3EIAT	

7.5.4.4.6.2.38.3

[Guideline] A Serving Endpoint is strongly recommended to include Wall Clock Time samples as defined in 7.5.4.4.3.15. If all of the following conditions are met:

- The RTSP PLAY request contains the WCT.dlna.org header (defined in guideline 7.5.4.4.6.2.18).
- The <val> parameter of the WCT.dlna.org header is 1.

[Attributes]

S	A	DMS +PU+	M-DMS	n/a	n/a	G45P7	

7.5.4.4.6.2.38.4

[Guideline] A Serving Endpoint shall not include Wall Clock Time samples as defined in 7.5.4.4.3.15 if all of the following conditions are met:

- The RTSP PLAY request contains the WCT.dlna.org header.
- The <val> parameter of the WCT.dlna.org header is 0.

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	n/a	MZ3MS	

-

7.5.4.4.6.2.39 MT RTP Seek Media Operation

7.5.4.4.6.2.39.1

[GUIDELINE] If a Receiving Endpoint supports the Seek Media Operation and the RTSP protocol state machine is in the Ready state, then it shall implement it by sending an RTSP PLAY request with the RTSP Range header.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[27]	33EIA	

NOTE: RTSP provides seeking capability by the inclusion of the RTSP Range header into the RTSP PLAY request. This enables random access within the RTSP session.

7.5.4.4.6.2.39.2

[GUIDELINE] If a Receiving Endpoint supports the Seek Media Operation and the RTSP protocol state machine is not in the Ready state, then it shall implement it by first bringing the RTSP protocol state machine into the Ready state, as described in guideline 7.5.4.4.6.2.36.2, and then by sending a PLAY request, as described in guideline 7.5.4.4.6.2.39.1.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	[27]	EIAT8	

7.5.4.4.6.2.39.3

[GUIDELINE] If the Serving Endpoint has specified a limited data range using the Available-Range.dlna.org header (guideline 7.5.4.4.6.2.16), then the values on the Range header in the PLAY request should be within the range specified by the Serving Endpoint.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[27]	IMZ3M	

NOTE: The Receiving Endpoint is discouraged from seeking outside the limited data range that was specified by the Serving Endpoint, even though it can do so

7.5.4.4.6.2.39.4

[GUIDELINE] A Serving Endpoint that supports the Seek Media Operation shall advertise this feature using the range SDP attribute (see guideline 7.5.4.4.6.2.64) and by setting the a-val parameter of the op-param field (defined in 7.4.1.3.18) or the lop-npt flag of the flags-param field (defined in 7.4.1.3.23) of the protocolInfo to 1. The Serving Endpoint shall also support the RTSP Range header in RTSP PLAY requests.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[27]	5P7QQ	C

7.5.4.4.6.2.39.5

[GUIDELINE] If a Serving Endpoint supports the Seek Media Operation, it shall implement the PAUSE method.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[27]	3MS6J	

7.5.4.4.6.2.39.6

[Guideline] RTP sequence numbers and RTP timestamps shall be continuous across jumps of NPT.

[Attributes]

M	C	DMS +PU+	M-DMS	n/a	[27]	Y2ZP8	

NOTE: As noted in guideline 7.5.4.4.6.2.37, the Serving Endpoint must include the Rtp-Info header with the seq parameter in the PLAY response, (if the value of the seq parameter is known at the time of issuing the response.)

7.5.4.4.6.2.40 MT RTP Receiving Endpoint Range header

7.5.4.4.6.2.40.1

[Guideline] If the Range header is included in a RTSP PLAY request, the header shall use "npt" range units. ("npt" range units are as per subclause 3.6 of RFC 2326.)

[Attributes]

M	L	DMP DMR	M-DMP	MIU	[27]	73UE8	

7.5.4.4.6.2.40.2

[Guideline] A Receiving Endpoint shall include no more than 1 Range header in the RTSP PLAY request.

[Attributes]

M	L	DMP DMR	M-DMP	MIU	[27]	QSPMX	

7.5.4.4.6.2.40.3

[Guideline] The Range header shall include exactly 1 range specifier.

[Attributes]

M	L	DMP DMR	M-DMP	MIU	[27]	YUFRH	

7.5.4.4.6.2.41 MT RTP Serving Endpoint Range header

7.5.4.4.6.2.41.1

[Guideline] A Serving Endpoint shall support the Range header in RTSP PLAY requests.

If the Range header requests a range that cannot be satisfied (for example, because the Serving Endpoint does not support the Seek Media Operation), the Serving Endpoint shall respond with status code 457 ("Invalid Range").

Note that if the Range header specifies a start time that is equal to the start time specified on the a=range attribute at the SDP session level (see guideline 1.189.4) the Range header shall be considered valid, and the Serving Endpoint shall not respond with status code 457 even if does not support the Seek Media Operation.

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	[27]	6W93G	

NOTE: This entry applies even if Serving Endpoint does not support the Seek Media Operation.

Example:

If the Serving Endpoint specifies "a=range:npt=0-" at the SDP session level, then the header "Range: npt=0-" is always valid, even if Serving Endpoint does not support the Seek Media Operation.

The Serving Endpoint that does not support the Seek Media Operation can simply do a string compare of the value specified on the Range header and the value indicated in SDP, and reject all other range strings with status code 457.

7.5.4.4.6.2.41.2

[Guideline] A Serving Endpoint shall include the Range header in the response to RTSP PLAY requests. The Range header shall use "npt" range units.

[Attributes]

M	L	DMS +PU+	M-DMS	n/a	[27]	V9K2V	

NOTE: This guideline applies even if Serving Endpoint does not support the Seek Media Operation.

7.5.4.4.6.2.41.3

[Guideline] If the Serving Endpoint is unable to determine the starting NPT time value of the content, such as for live captured content, then the npt range units may be specified as "now-".

[Attributes]

O	C	DMS +PU+	M-DMS	n/a	[27]	S9MDT	

NOTE: When streaming live content, the Serving Endpoint might be unable to provide numerical values for the Range header.

7.5.4.4.6.2.42 MT RTP Scale header

[Guideline] If a Scale header is included in the PLAY request then the Serving Endpoint shall indicate the scale value that it chose using a Scale header in the PLAY response.

The RTP Serving Endpoint shall support the Scale header in a RTSP PLAY request if the value of the Scale header is numerically identical to the one of the scale-param values in the scale-value token. Furthermore the RTP Serving Endpoint shall support all scale-value tokens that are listed in the ps-param of the 4th field protocolInfo. The RTSP Scale header is defined in subclause 12.34 of RFC-2326 ([27]).

[Attributes]

M	C	DMS +PU+	M-DMS	n/a	[27]	HHKDS	

NOTE: If Receiving Endpoint uses a scale value that's not included in the server-speed parameter of the maxsp-param (defined in 7.4.1.3.25) in the 4th field of the protocolInfo then Serving Endpoint can choose an alternate scale value at its discretion.

7.5.4.4.6.2.43 MT RTP Scan Media Operations

7.5.4.4.6.2.43.1

[General] Fast Forward Scan, Slow Forward Scan, Fast Backward Scan, Slow Backward Scan

7.5.4.4.6.2.43.2

[Guideline] Serving Endpoints and Receiving Endpoints that implement the RTP Media Transport may support any of the Scan Media Operations

[ATTRIBUTES]

O	A	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	[27]	JY2ZP	

7.5.4.4.6.2.43.3

[GUIDELINE] A Receiving Endpoint should use the Scale header of the PLAY method to signal any of the Scan Media Operations.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[27]	O73UE	

NOTE: The Scale header is recommended over the Speed header, because Scale allows the Fast Forward Scan Media Operation to occur using less bandwidth than with the Speed header.

7.5.4.4.6.2.43.4

[GUIDELINE] A Receiving Endpoint may use the Speed header of the PLAY method to signal a Fast Forward Scan or Slow Forward Scan media operation.

[ATTRIBUTES]

O	R	DMP DMR	M-DMP	MIU	[27]	6QSPM	

7.5.4.4.6.2.44 MT RTP Serving Endpoint Scan Media Operations support

7.5.4.4.6.2.44.1

[GUIDELINE] If Scan Media Operations are supported, the Serving Endpoint shall support the Scale and Range headers of the PLAY method.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[27]	YYUFR	

7.5.4.4.6.2.44.2

[GUIDELINE] If Scan Media Operations are supported, the Serving Endpoint shall implement the PAUSE method.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[27]	46W93	

7.5.4.4.6.2.45 MT RTP Speed header support

7.5.4.4.6.2.45.1

[GUIDELINE] A Serving Endpoint may support the Speed header of the PLAY method.

[ATTRIBUTES]

O	R	DMS +PU+	M-DMS	n/a	[27]	YV9K2	

7.5.4.4.6.2.45.2

[GUIDELINE] If a Speed header is included in the PLAY request then the Serving Endpoint shall indicate the speed value that it chose using a Speed header in the PLAY response

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[27]	DS9MD

NOTE: If Receiving Endpoint uses a speed value that exceeds the attribute value of the maxsp-param (7.4.1.3.25 MM maxsp-param (Maximum RTSP Speed header value))in the 4th field of the protocolInfo then Serving Endpoint can choose an alternate Speed value at its discretion

7.5.4.4.6.2.45.3

[GUIDELINE] If the Receiving Endpoint uses a value of the Speed header that's not included in the 4th field of the protocolInfo then the Serving Endpoint may choose an alternate value of the Speed header.

[ATTRIBUTES]

O	C	DMS +PU+	M-DMS	n/a	[27]	WYYUF

7.5.4.4.6.2.45.4

[GUIDELINE] A Receiving Endpoint shall be prepared to receive a different value of the Speed header in the PLAY response than what it requested in the PLAY request.

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[27]	Q6QSP

7.5.4.4.6.2.45.5

[GUIDELINE] If Receiving Endpoint specifies the Speed header in a PLAY request, its value shall be greater than zero.

[ATTRIBUTES]

M	R	DMP DMR	M-DMP	MIU	[27]	ZO73U

NOTE: The guideline clarifies that the Speed Header value cannot be zero or negative.

7.5.4.4.6.2.46 MT RTP Buffer Parameters in PLAY response

7.5.4.4.6.2.46.1

[GUIDELINE] The Serving Endpoint may indicate updated buffer parameters required for playback of the media stream by using following RTSP headers in the response to the PLAY request:

- predec-buffer-size-line = "Predec-Buffer-Size.dlna.org" *LWS ":" *LWS url-size-pair *(*LWS "," *LWS url-size-pair)
- url-size-pair = "url" "=" stream-url ";" "size" "=" size-val
- stream-url = <case sensitive, HTTP-escaped URL string, with a maximum length of 1024 bytes in its UTF-8 encoded form>
- size-val = value
- value = 1*DIGIT

Note that the literals, "url" and "size" are case sensitive.

The size-val parameter specifies the minimum pre-decoder buffer size, in bytes, required for the RTP stream identified by stream-url for the specified play speed/scale setting.

- initial-buffering-line = "Initial-Buffering.dlna.org" *LWS ":" *LWS "url-time-pair *(*LWS "," *LWS url-time-pair)

- url-time-pair = "url" "=" stream-url ";" "time" "=" time-val
- stream-url = stream-url = <case sensitive, HTTP-escaped URL string, with a maximum length of 1024 bytes in its UTF-8 encoded form>
- time-val = value
- value = 1*DIGIT

Note that the literals, "url" and "time", are case sensitive.

The time-val parameter specifies the minimum initial buffering in NPT milliseconds the Receiving Endpoint may use for the media stream identified by stream-url. NPT and its relationship to units of time are defined in subclause 3.6 of [27].

stream-url is the URL for the RTP stream that the value tokens apply to. It may be either an absolute URL or a relative URL. (See RFC-2326, subclause C.1.1, for rules for how to convert a relative URL to an absolute URL.)

Example:

- Predec-Buffer-Size.dlna.org: url=audio;size=11200, url=video;size=243250 Initial-Buffering.dlna.org: url=audio;time=500, url=video;time=1000

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	[27]	OJY2Z	

NOTE: The Serving Endpoint can specify the minimum pre-decoder buffer size required to avoid pre-decoder buffer overflow and underflow when receiving an RTP stream as a result of RTSP PLAY request.

If the values of the RTSP headers Speed and Scale are both 1, then Pre-Decoder Buffer Size will be less than or equal to the value of the a=predecbufsize.dlna.org attribute signaled in the SDP. For Speed or Scale values not equal to 1, Pre-Decoder Buffer Size can be larger than the value of the a=predecbufsize.dlna.org attribute in SDP.

The NPT time can be derived from the decode time of the RTP payload.

7.5.4.4.6.2.46.2

[GUIDELINE] If the Predec-Buffer-Size.dlna.org header is specified in the response to a PLAY request, then the Receiving Endpoint should use a pre-decoder buffer that is at least as large as the value specified by this header in order to avoid decoder underflow during Streaming Transfer.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	n/a	346W9	

NOTE: If the pre-decoder buffer is combined with the network de-jitter buffer, this guideline refers to the size of the combined buffer.

7.5.4.4.6.2.47 MT RTP Pause Media Operation

7.5.4.4.6.2.47.1

[GUIDELINE] If the Serving Endpoint and the Receiving Endpoint support the Pause Media Operation, then the Receiving Endpoint should implement it by sending an RTSP PAUSE request.

[ATTRIBUTES]

S	C	DMP DMR	M-DMP	MIU	[27]	OYV9K	

NOTE: RTSP PAUSE request simply stops the RTP media data transport, but keeps the RTSP session alive for future requests.

Serving Endpoint indicates support for Pause media operation by setting the rtsp-pause flag in the DLNA.ORG_FLAGS param (defined in 7.4.1.3.23 MM flags-param (Flags Parameter)) in the 4th field of the protocolInfo.

7.5.4.4.6.2.47.2

[Guideline] If the response to a PAUSE request indicates error code 455 ("Method Invalid in the current state"), then this error should be ignored by the Receiving Endpoint.

[Attributes]

O	A	DMP DMR	M-DMP	MIU	[27]	BOJY2	

NOTE: Some serving endpoints can enter Ready state at the end of the RTP stream. In that case they might respond to a PAUSE request with a 455 error. This can be safely ignored by the Receiving Endpoint.

7.5.4.4.6.2.47.3

[Guideline] When a RTSP session is paused, then the Serving Endpoint shall stop transmiting RTP packets for all RTP streams in the RTSP session.

[Attributes]

M	C	DMS +PU+	M-DMS	n/a	[27] [38]	JZO73	

NOTE: It is not acceptable to force the Receiving Endpoint to drop packets.

7.5.4.4.6.2.48 MT RTP PAUSE requests with a Range header.

7.5.4.4.6.2.48.1

[Guideline] A Receiving Endpoint should not include the Range header in a PAUSE request.

[Attributes]

S	L	DMP DMR	M-DMP	MIU	[27]	QQ6QS	

NOTE: The Range header in the PLAY request can be used to accomplish a similar function as a Range header in a PAUSE request.

7.5.4.4.6.2.48.2

[Guideline] A Serving Endpoint may respond to a PAUSE request that contains a Range header with error 457 ("Invalid Range").

[Attributes]

O	C	DMS +PU+	M-DMS	n/a	[27]	ZWYYU	

NOTE: Serving Endpoints are not required to implement "queued PAUSE".

7.5.4.4.6.2.49 MT RTP Pause-Release Media Operation

[Guideline] If the Serving Endpoint and the Receiving Endpoint both support the Pause-Release Media Operation, then the Receiving Endpoint shall implement the Pause Media Operation by sending a PLAY request. The PLAY request shall not include the Range header.

[Attributes]

M	A	DMP DMR	M-DMP	MIU	[27]	R346W	C

NOTE: Serving Endpoint indicates support for Pause media operation by setting the rtsp-pause flag in the DLNA.ORG_FLAGS param (defined in 7.4.1.3.23 MM flags-param (Flags Parameter)) in the 4th field of the protocolInfo.

7.5.4.4.6.2.50 MT RTP time stamp clock is not paused while in RTSP Paused state

7.5.4.4.6.2.50.1

[Guideline] If a Serving Endpoint transitions from RTSP Paused state into the RTSP Playing state, then the RTP time stamps shall be incremented by the amount of NPT (measured in RTP time stamp units) that the Serving Endpoint spent in RTSP Paused state.

[Attributes]

M	C	DMS +PU+	M-DMS	n/a	[27]	SBOJY	

NOTE: The RTP time stamp "clock" is not paused when the Serving Endpoint is in Paused state. After a Pause-Release or Seek Media operation, the RTP time stamps must be increased by an amount corresponding to the length of time that the server was paused.

7.5.4.4.6.2.50.2

[Guideline] If a Serving Endpoint transitions from RTSP Paused state into RTSP Playing state, then it shall support that the RTP time stamps of the RTP packets received after entering Playing state may not be continous with time stamps of RTP packets received prior to entering Paused state.

[Attributes]

M	C	DMP DMR	M-DMP	MIU	[27]	6JZO7	

NOTE: A Receiving Endpoint might need to recompute the mapping between NPT time and RTP timestamp each time it receives a PLAY response.

7.5.4.4.6.2.51 MT RTP End Of Stream indication support

[Guideline] If a Receiving Endpoint wants to receive an ANNOUNCE request when the Serving Endpoint has reached the end of the requested play range, then the Receiving Endpoint shall include the Supported header with the feature-tag dlna.announce in the PLAY request.

Example:

- Supported: dlna.announce

[Attributes]

M	A	DMP DMR	M-DMP	MIU	[27]	7QQ6Q	

7.5.4.4.6.2.52 MT RTP End Of Stream indication

7.5.4.4.6.2.52.1

[Guideline] If a PLAY request includes the Supported header with the feature tag dlna.announce, then the Serving Endpoint should send an ANNOUNCE request to the Receiving Endpoint when the last RTP packet has been sent.

[Attributes]

S	A	DMS +PU+	M-DMS	n/a	[27]	6ZWYY	

NOTE: The purpose of the ANNOUNCE request is to indicate that Serving Endpoint has reached the end of the requested play range, and to indicate the last RTP sequence number for each RTP stream.

7.5.4.4.6.2.52.2

[GUIDELINE] The ANNOUNCE request for the purpose of sending an end of stream indication shall include the following RTSP headers:

- Session
- CSeq
- Rtp-Info
- Event-Type.dlna.org

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[27]	8SBOJ

7.5.4.4.6.2.52.3

[GUIDELINE] The Rtp-Info header shall specify the seq parameter for each selected RTP stream. The value of the seq parameter shall be equal to the RTP sequence number of the last RTP packet plus one.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[27]	S6JZO

NOTE: The Rtp-Info header indicates the "next" RTP sequence number for each RTP stream. The "next" sequence number is the same as the last sequence number plus one.

7.5.4.4.6.2.52.4

[GUIDELINE] The Event-Type.dlna.org header shall specify the event-code 2000.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	P7QQ6

NOTE: Event code 2000 indicates that Serving Endpoint has reached the end of the requested play range.

7.5.4.4.6.2.53 MT RTP Current Limited Data Range indication

7.5.4.4.6.2.53.1

[GUIDELINE] A Serving Endpoint may send an ANNOUNCE request to indicate the current limited data range (seekable range) if all of the following conditions are met:

- A PLAY request included the Supported header with the feature tag dlna.announce
- The Available-Range.dlna.org header was included in the DESCRIBE response
- The current limited seekable range, which is defined as the closed range bounded by the UCDAM's data positions of r_0 and r_N, has changed since the last time Serving Endpoint included the Available-Range.dlna.org header in any of the following RTSP methods: An ANNOUNCE request to indicate the current limited seekable range, a DESCRIBE response, an OPTIONS response
- Serving Endpoint has not sent an ANNOUNCE request to indicate the current limited seekable range in the last N seconds. (The parameter N is defined in guidelines 7.5.4.4.6.2.53.5 and 7.5.4.4.6.2.53.6.)

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	[27]	T8SBO

NOTE: See guideline 7.4.1.3.28 for a definition of limited data range.

7.5.4.4.6.2.53.2

[GUIDELINE] If a Serving Endpoint sends an ANNOUNCE request to indicate the current limited seekable range, then the request shall include the following RTSP headers:

- Session
- CSeq
- Event-Type.dlna.org (guideline 7.5.4.4.6.2.15)
- Available-Range.dlna.org (guideline 7.5.4.4.6.2.16)

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	AT8SB	

7.5.4.4.6.2.53.3

[GUIDELINE] If the Available-Range.dlna.org header is included in an ANNOUNCE request, the parameters on that header should be set as described in guideline 7.5.4.4.6.2.26.2.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	n/a	MS6JZ	

7.5.4.4.6.2.53.4

[GUIDELINE] If a Serving Endpoint sends an ANNOUNCE request to indicate the current limited seekable range, the Event-Type.dlna.org header in that request shall specify the event-code 9000.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	PMX34	

Event code 9000 indicates that the purpose of the ANNOUNCE request is to indicate the current limited seekable range.

7.5.4.4.6.2.53.5

[GUIDELINE] The recommended value for N in guideline 7.5.4.4.6.2.53.1 is 20.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	n/a	FRHZG	

7.5.4.4.6.2.53.6

[GUIDELINE] The value for N in guideline 7.5.4.4.6.2.53.1 shall not be less than 1.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	93GF2	

7.5.4.4.6.2.54 MT RTP OPTIONS request

7.5.4.4.6.2.54.1

[GUIDELINE] A Serving Endpoint shall support the Session header in an OPTIONS request.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[27]	K2V43	

NOTE: A Receiving Endpoint can send an OPTIONS request with a Session header as a way to keep the RTSP session alive, or to query the current limited seekable range.

7.5.4.4.6.2.54.2

[GUIDELINE] If the Session header in an OPTIONS request identifies an RTSP session for which the Serving Endpoint supports the Seek Media Operation, and if the Serving Endpoint supports the Limited Random Access Data Availability model this content binary (see guideline 7.4.1.3.28) for the RTSP session, then the Available-Range.dlna.org header (guideline 7.5.4.4.6.2.16) shall be included in the OPTIONS response.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[27]	MDTQF	

7.5.4.4.6.2.54.3

[GUIDELINE] If the Available-Range.dlna.org header is included in an OPTIONS response, the parameters on that header should be set as described in guideline 7.5.4.4.6.2.26.2.

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	n/a	KDS6Y	

7.5.4.4.6.2.54.4

[GUIDELINE] If the DESCRIBE response included the Available-Range.dlna.org header, then the Receiving Endpoint may query the Serving Endpoint about the current limited seekable range by sending an OPTIONS request with the Session header.

[ATTRIBUTES]

O	A	DMP DMR	M-DMP	MIU	n/a	TVQUZ	

7.5.4.4.6.2.55 MT RTP Stop Media Operation

7.5.4.4.6.2.55.1

[GUIDELINE] A Receiving Endpoint shall implement the Stop Media Operation by sending a RTSP TEARDOWN request for the aggregate control URI.

[ATTRIBUTES]

M	L	DMP DMR	M-DMP	MIU	[27]	5Y4A5	

NOTE: The RTSP TEARDOWN request ends the RTSP session, but the same TCP connection can be used to start another RTSP session.

7.5.4.4.6.2.55.2

[GUIDELINE] A Receiving Endpoint should wait for the response from the Serving Endpoint after sending a TEARDOWN request.

[ATTRIBUTES]

S	C	DMP DMR	M-DMP	MIU	[27]	3TVQU	

7.5.4.4.6.2.56 MT RTP SDP Character Set

[GUIDELINE] The SDP description shall be encoded in the ISO 10646 character set in UTF-8.

[ATTRIBUTES]

M	R	DMS +PU+	M-DMS	n/a	[28]	HKDS6

NOTE: RTSP also uses this character set.

7.5.4.4.6.2.57 MT RTP SDP Case Sensitivity

[GUIDELINE] The SDP protocol is a case-sensitive protocol.

[ATTRIBUTES]

M	A	DMS DMP DMR DMC DMPr	M-DMS M-DMP M-DMC M-DMD M-DMU	MIU	[28]	9MDTQ

7.5.4.4.6.2.58 MT RTP SDP Optional Values

[GUIDELINE] A Receiving Endpoint shall ignore (or tolerate) any SDP attributes that it does not support.

[ATTRIBUTES]

M	R	DMP DMR	M-DMP	MIU	[28]	UFRHZ

7.5.4.4.6.2.59 MT RTP SDP field order

7.5.4.4.6.2.59.1

[GUIDELINE] SDP fields shall be specified in the order defined by RFC-2327, Appendix A.

[ATTRIBUTES]

M	R	DMS +PU+	M-DMS	n/a	[28]	9K2V4

7.5.4.4.6.2.59.2

[GUIDELINE] If SDP fields are not in the defined order, the Receiving Endpoint should accept and process those fields.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	[28]	W93GF

7.5.4.4.6.2.60 MT RTP SDP Session Description Fields

7.5.4.4.6.2.60.1

[GUIDELINE] The following SDP fields are required for the SDP session description section:

- version field (v=)
- origin field (o=)
- session name field (s=)
- time field (t=)
- control URL attribute field (a=control)
- range attribute field (a=range)
- DLNA contentFeatures field (a=contentFeatures.dlna.org)

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	[27] [28]	SPMX3	

7.5.4.4.6.2.60.2

[Guideline] The following SDP fields are optional for the SDP session description section:

- Bandwidth modifier field (b=)
- Connection field (c=)
- DLNA scmsFlag field (a=scmsFlag.dlna.org)

[Attributes]

O	A	DMS +PU+	M-DMS	n/a	[28]	3UE86	

7.5.4.4.6.2.60.3

[Guideline] If the connection field is not specified in the SDP session description section, then it shall be specified in each SDP media description section.

[Attributes]

M	R	DMS +PU+	M-DMS	n/a	[28]	GF288	

7.5.4.4.6.2.61 MT RTP Contents of SDP Origin Field

7.5.4.4.6.2.61.1

[Guideline] The SDP origin field (o=) shall have the following format:

- sdp-origin-field = "o=" <username> SP <session id> SP <version> SP <network type> <address type> SP <address>

The literal, "o=", is case sensitive.

<username>, <session id>, <network type>, <address type>, and <address> shall be set in accordance to RFC-2327 with the exceptions mentioned in the next guideline entry.

[Attributes]

M	L	DMS +PU+	M-DMS	n/a	[28]	V43RH	

7.5.4.4.6.2.61.2

[Guideline] Allowed exceptions:

- <username> may be set to "-".
- <address> may be unspecified ("0.0.0.0" when IPv4 is used).

[Attributes]

O	L	DMS +PU+	M-DMS	n/a	n/a	TQF9O	

NOTE: Recommendations address privacy concerns.

7.5.4.4.6.2.62 MT RTP Contents of SDP Session Name Field

7.5.4.4.6.2.62.1

[Guideline] The SDP session name field (s=) shall be present. It has the following format:

- sdp-session-name-field = "s="<session name>

Note that the literal, "s=", is case sensitive.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[28]	S6YJS	

7.5.4.4.6.2.62.2

[GUIDELINE] <session name> may be set to a single space character (' ') when session name is not available.

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	n/a	QUZYZ	

7.5.4.4.6.2.63 MT RTP SDP control attribute

7.5.4.4.6.2.63.1

[GUIDELINE] The a=control attribute shall refer to a resource on the Serving Endpoint. The scheme of the URL in the control attribute, if any, shall be "rtsp". The host element, if any, of the URL in the a=control attribute shall belong to the Serving Endpoint. This URL must be a case sensitive HTTP-escaped string, with a maximum length of 1024 bytes in its UTF-8 encoded form.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[27]	4A5ZG	C

NOTE: It is not allowed to use a=control as a way to redirect the Receiving Endpoint to a different Serving Endpoint or as a way to replace RTSP with another protocol.

Subsection 3.2 of RFC-2326 defines the syntax of RTSP URLs.

7.5.4.4.6.2.63.2

[GUIDELINE] Receiving Endpoint shall support the use of relative URLs in the a=control attribute.

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[27]	NQCR9	

NOTE: The following are examples of a=control attributes that specify relative URLs:

- a=control:stream=1
- a=control:*

The usage of relative URLs is described in subclause C.1.1 of RFC-2326.

7.5.4.4.6.2.64 MT RTP SDP range attribute at the SDP session level

7.5.4.4.6.2.64.1

[GUIDELINE] The a=range attribute at the SDP session level shall use "npt" range units.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[27]	24UR4	

NOTE: "Npt" range units are defined in subclause 3.6 of RFC 2326.

7.5.4.4.6.2.64.2

[GUIDELINE] If a Serving Endpoint supports the Full Random Access Data Availability model for a content binary, then the seekable range must be indicated using the a=range attribute at the SDP session level. In this case, the a=range attribute shall include both a start time and a stop time.

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[27]	LNQCR	C

NOTE: The a=range attribute specifies the start and stop time only if the seekable range never changes.

7.5.4.4.6.2.64.3

[GUIDELINE] If a Serving Endpoint supports Limited Random Access Data Availability model for a content binary, then the a=range attribute at the SDP session level shall specify an open-ended interval. The start time shall be specified as a value that falls within the current seekable range, defined as the closed range bounded by the UCDAM's data positions of r_0 and r_N, or as "now".

Examples:

- a=range:npt=now-
- a=range:npt=0-

[ATTRIBUTES]

M	L	DMS +PU+	M-DMS	n/a	[27]	Y4A5Z	C

7.5.4.4.6.2.65 MT RTP SDP scmsFlag.dlna.org attribute

[GUIDELINE] If used, the format of the scmsFlag.dlna.org SDP attribute field shall comply with the following syntax:

- scmsFlag.dlna.org = "a=scmsFlag.dlna.org:" flagValue
- With syntax and symantics of flagValue as defined in guideline 7.5.4.3.2.8 of HTTP MT.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	VQUZY	

7.5.4.4.6.2.66 MT RTP SDP contentFeatures.dlna.org attribute

[GUIDELINE] The format of the contentFeatures.dlna.org SDP attribute field shall be as follows:

- contentFeatures.dlna.org = "contentFeatures.dlna.org:" 4th_field
- Please note that 4th_field is defined in guideline 7.4.1.3.16.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	DS6YJ	E

NOTE: The 4th_field is the 4th field of the protocolInfo value supplied by the serving endpoint for this content binary as defined in guideline 7.4.1.3.16.

7.5.4.4.6.2.67 MT RTP SDP Media Description Fields

7.5.4.4.6.2.67.1

[GUIDELINE] The following SDP fields are required for each media description:

- media field ('m=')
- control URL attribute field (a=control)

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[27] [28]	DTQF9

7.5.4.4.6.2.67.2

[GUIDELINE] The following SDP fields are optional for each media description:

- rtpmap attribute field (a=rtpmap) Only required if conditions described in guideline 7.5.4.4.6.2.69 apply.
- fmtp attribute field (a=fmtp) Only required if mandated by the Media Format Profile.
- Bandwidth modifier field (b=)
- connection field (c=)
- Range attribute field (a=range)
- Acceptable RTP/AVPF feedback message types (a=rtcp-fb)
- DLNA pre-decoder buffer size (a=predecbufsize.dlna.org)
- DLNA minimum required pre-decoder buffer size (a=adaptation-predecbufsize.dlna.org)
- DLNA transmission rate adaptation field (a=trans-rate-adapt.dlna.org)

[ATTRIBUTES]

O	A	DMS +PU+	M-DMS	n/a	[27] [28]	2V43R

7.5.4.4.6.2.68 MT RTP Contents of SDP Media Field

[GUIDELINE] The SDP media field (m=) shall have the following format:

- sdp-media-field = "m=" media SP port SP transport SP payload-format
- port = "0"
- transport = "RTP/AVP" | "RTP/AVPF"
- payload-format = 1*DIGIT; valid RTP payload type numbers.

Note that the literals, "m=", "RTP/AVP", and "RTP/AVPF", are case sensitive.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[27] [28]	3GF28

NOTE: <port> must be zero because the Receiving Endpoint will select a different port using the RTSP SETUP method.

7.5.4.4.6.2.69 MT RTP SDP Rtpmap Attribute Field

[GUIDELINE] The SDP rtpmap attribute is required for RTP dynamic payload types, or for static payload types if a non-standard clock speed or other RTP parameters need to be communicated to the Receiving Endpoint.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[28]	RHZG5

NOTE: See clause 6 of RFC 2327 for an example on using the rtpmap attribute.

7.5.4.4.6.2.70 MT RTP SDP range attribute at the SDP media level

7.5.4.4.6.2.70.1

[Guideline] If the start and stop times of all media streams are readily available, and all media streams do not have an identical start time, or do not have an identical stop time, then the Serving Endpoint is strongly recommended to include the a=range attribute at the SDP media level for each media stream.

[Attributes]

S	A	DMS +PU+	M-DMS	n/a	[27]	43RHS

NOTE: If media streams are stored in a file which is compliant with the file format defined in 3GPP PSS [94], then the start and stop times of each media stream are usually readily available as fields in the file format.

7.5.4.4.6.2.70.2

[Guideline] The a=range attribute at the SDP media level shall use "npt" range units.

[Attributes]

M	C	DMS +PU+	M-DMS	n/a	[27]	6GKZQ

7.5.4.4.6.2.70.3

[Guideline] The a=range attributes at the SDP media level shall all be subranges of the range indicated by the a=range attribute at the SDP session level.

[Attributes]

M	C	DMS +PU+	M-DMS	n/a	[27]	F9OV4

7.5.4.4.6.2.71 MT RTP SDP b= field

7.5.4.4.6.2.71.1

[Guideline] A Serving Endpoint should include a SDP b= field with the AS bandwidth modifier for each RTP stream.

[Attributes]

S	L	DMS +PU+	M-DMS	n/a	[28]	YJSWA

NOTE: This bandwidth modifier specifies the maximum bit rate of the RTP stream.

7.5.4.4.6.2.71.2

[Guideline] A Serving Endpoint should include a SDP b= field with the RR bandwidth modifier for each RTP stream.

[Attributes]

S	L	DMS +PU+	M-DMS	n/a	[40]	ZYZLT

NOTE: This bandwidth modifier allows the Serving Endpoint to specify the bit rate used for RTCP Receiver Reports.

7.5.4.4.6.2.71.3

[Guideline] A Serving Endpoint should include a SDP b= field with the RS bandwidth modifier for each RTP stream.

[ATTRIBUTES]

S	L	DMS +PU+	M-DMS	n/a	[40]	5ZGPK	

7.5.4.4.6.2.71.4

[GUIDELINE] A Receiving Endpoint shall support the AS bandwidth modifier for the SDP b= field.

[ATTRIBUTES]

M	L	DMP DMR	M-DMP	MIU	[28]	CR9YS	

NOTE: The bit rate allowed for RTCP Receiver Reports is 2.5% of the value on the b=AS field, unless overridden by a b=RR field.

7.5.4.4.6.2.71.5

[GUIDELINE] A Receiving Endpoint shall support the RR bandwidth modifier for the SDP b= field.

[ATTRIBUTES]

M	L	DMP DMR	M-DMP	MIU	[40]	UR4XV	

NOTE: b=RR:0 means that Receiving Endpoint must not send any RTCP Receiver Reports.

7.5.4.4.6.2.72 MT RTP/AVPF support in SDP

7.5.4.4.6.2.72.1

[GUIDELINE] A Serving Endpoint may use the RTP/AVPF profile for RTCP-based feedback in one or more of the media descriptions in SDP.

[ATTRIBUTES]

O	C	DMS +PU+	M-DMS	n/a	[8]	ZLTOT	

7.5.4.4.6.2.72.2

[GUIDELINE] If the Serving Endpoint uses the RTP/AVPF profile for an RTP stream, this shall be indicated by setting the <transport> field in the media field to "RTP/AVPF" and by using the a=rtcp-fb attribute to specify the permitted RTCP feedback messages types.

[ATTRIBUTES]

M	C	DMS +PU+	M-DMS	n/a	[8]	GPK9T	

NOTE: The following is a sample media description, which indicates support for RTP/AVPF and which specifies that the nack feedback message type is supported:

- m=audio 0 RTP/AVPF 14
- a=rtcp-fb:14 nack

7.5.4.4.6.2.73 MT RTP Tolerate RTP/AVPF in SDP

[GUIDELINE] A Receiving Endpoint shall tolerate media descriptions that specify the RTP/AVPF profile. If a Receiving Endpoint does not support the RTP/AVPF profile, it shall treat the media description as if it specified the RTP/AVP profile.

[ATTRIBUTES]

M	C	DMP DMR	M-DMP	MIU	[8]	9YSY6	

7.5.4.4.6.2.74 MT RTP/AVPF nack feedback message type in SDP

[GUIDELINE] A Receiving Endpoint should support the nack RTCP feedback message type in the RTP/AVPF profile.

[ATTRIBUTES]

S	L	DMP DMR	M-DMP	MIU	[8]	4XV7O	

7.5.4.4.6.2.75 MT RTP bfr feedback message type in SDP

[GUIDELINE] A Receiving Endpoint should support the bfr RTCP feedback message type.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	n/a	SY6VW	

NOTE: Guideline 7.5.4.4.4.15.3 describes the syntax of the bfr RTCP feedback message.

7.5.4.4.6.2.76 MT RTP Buffer Fullness Support indication in SDP

7.5.4.4.6.2.76.1

[GUIDELINE] If the Serving Endpoint expects to receive Buffer Fullness Reports (BFR's) from the Receiving Endpoint, it shall use the RTP/AVPF profile for the RTP stream and it shall specify bfr as one of the acceptable feedback message types, as described in guideline 7.5.4.4.6.2.72.2.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	V7O96	

7.5.4.4.6.2.76.2

[GUIDELINE] When the rtcp-fb-id parameter of the rtcp-fb attribute is set to "bfr", the corresponding rtcp-fb-param parameter shall be set to a byte string representation of the maximum interval between BFR's sent from the Receiving Endpoint to the Serving Endpoint.

Example:

- a=rtcp-fb:96 bfr 500

In this example, "500" indicates that a BFR shall be sent at least once every 500 ms.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	[8]	7O96E	

NOTE: rtcp-fb-id and rtcp-fb-param" are defined in [8].

7.5.4.4.6.2.77 MT RTP Support trr-int SDP parameter if RTP/AVPF is supported

[GUIDELINE] A Receiving Endpoint which supports the RTP/AVPF profile shall implement support for the trr-int parameter in the a=rtcp-fb attribute.

[ATTRIBUTES]

M	L	DMP DMR	M-DMP	MIU	[8]	PK9TG	

NOTE: The trr-int parameter allows the Serving Endpoint to specify a minimal time interval between full (complete) RTCP Receiver Reports.

7.5.4.4.6.2.78 MT RTP Pre-decoder buffer size indication in SDP

7.5.4.4.6.2.78.1

[GUIDELINE] The Serving Endpoint should indicate the minimum pre-decoder buffer size that is required for the media stream using the following SDP media-level attribute:

- predecbufsize-attribute = "a=predecbufsize.dlna.org:" value
- value = 1*DIGIT

Note that the literal, "a=predecbufsize.dlna.org:", is case sensitive.

The value token specifies the required minimum pre-decoder buffer size in bytes for the media stream that the Receiving Endpoint shall have for normal speed playback (speed=1, scale=1).

Example:

- a=predecbufsize.dlna.org:480750

[ATTRIBUTES]

S	A	DMS +PU+	M-DMS	n/a	n/a	YSY6V	

NOTE: The Serving Endpoint can specify the required minimum pre-decoder buffer size for the media stream to avoid the Receiving Endpoint (pre-decoder) buffer underflow.

7.5.4.4.6.2.78.2

[GUIDELINE] The attribute predecbufsize.dlna.org shall not exceed the size of the pre-decoder buffer defined in the media format profile used.

[ATTRIBUTES]

M	A	DMS +PU+	M-DMS	n/a	n/a	XV7O9	

NOTE: DLNA Media format profile refers to underlying standards (codecs or system layer) which in turn define the size of pre-decoder buffer.

7.5.4.4.6.2.78.3

[GUIDELINE] If the a=predecbufsize.dlna.org SDP attribute is specified, then the Receiving Endpoint should use a pre-decoder buffer that is at least as large as the value specified by this attribute in order to avoid decoder underflow.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	n/a	QF9OV	

7.5.4.4.6.2.79 MT RTP Minimum required pre-decoder buffer size indication in SDP

7.5.4.4.6.2.79.1

[GUIDELINE] The Serving Endpoint that supports encoding rate adaptation mechanisms may indicate absolute minimum required pre-decoder buffer size for the media stream based on its encoding rate adaptation capabilities using the following SDP media-level attribute:

- adaptation-predecbufsize-attribute = "a=adaptation-predecbufsize.dlna.org:" value
- value = 1*DIGIT

Note that the literal, "a=adaptation-predecbufsize.dlna.org:", is case sensitive.

This specifies the absolute minimum pre-decoder buffer size in bytes that the Receiving Endpoint shall have to be able to render this media stream.

Example:

- a=adaptation-predecbufsize.dlna.org:84752

[Attributes]

O	A	DMS +PU+	M-DMS	n/a	n/a	6YJSW	

NOTE: The Serving Endpoint can support adapting the encoding rate of the media stream to support a pre-decoder buffer size at the Receiving Endpoint that is smaller than the minimum required pre-decoder buffer size. The absolute minimum pre-decoder buffer size when the encoding rate is adapted, can be signaled by the Serving Endpoint using this SDP attribute.

Examples of encoding rate adaptation mechanisms are trans-rating, trans-coding and live encoding at different bit rates.

7.5.4.4.6.2.79.2

[Guideline] exceed the value of the attribute predecbufsize.dlna.org.

[Attributes]

M	L	DMS +PU+	M-DMS	n/a	n/a	UZYZL	

7.5.4.4.6.2.79.3

[Guideline] If the a=adaptation-predecbufsize.dlna.org SDP attribute is specified then the Receiving Endpoint should use a pre-decoder buffer of a size that is at least as large as the value specified by this attribute.

[Attributes]

S	A	DMP DMR	M-DMP	MIU	n/a	A5ZGP	

7.5.4.4.6.2.80 MT RTP SDP transmission rate adaptation field

7.5.4.4.6.2.80.1

[Guideline] If the Serving Endpoint supports adapting the transmission rate of the RTP stream without also adapting the encoding rate by the same amount, then it shall include the a=trans-rate-adapt.dlna.org:1 attribute at the SDP media-level. The syntax for the a=trans-rate-adapt.dlna.org attribute is as follows:

- trans-rate-adapt-attribute = "a=trans-rate-adapt.dlna.org:" bin
- bin = "0" | "1"

Note that the literal, "a=trans-rate-adapt.dlna.org:", is case sensitive.

The bin token is 0 if no transmission rate adaptation will be performed, 1 if transmission rate adaptation is possible.

Note that even if the a=trans-rate-adapt.dlna.org SDP attribute indicates that transmission rate adaptation is possible, the Serving Endpoint is not allowed to perform transmission rate adaptation unless additional requirements are satisfied (see guideline 7.5.4.4.3.14).

[Attributes]

M	A	DMS +PU+	M-DMS	n/a	n/a	QCR9Y	

NOTE: If the Serving Endpoint supports transmission rate adaptation, i.e., speed up or slow down the transmission rate of the RTP packets, it must indicate this by specifying 1 on the <bin> field of the a=trans-rate-adapt.dlna.org SDP attribute.

7.5.4.4.6.2.80.2

[GUIDELINE] If the Receiving Endpoint is transmitting Buffer Fullness Reports and the <bin> field of the a=trans-rate-adapt.dlna.org attribute is equal to 1, then the Receiving Endpoint shall decide the rate at which content is presented.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	n/a	4UR4X	

7.5.4.4.6.2.80.3

[GUIDELINE] If the Receiving Endpoint is transmitting Buffer Fullness Reports and the <bin> field of the a=trans-rate-adapt.dlna.org attribute is equal to 1, then a Receiving Endpoint is recommended not to recover Serving Endpoints clock.

[ATTRIBUTES]

S	A	DMP DMR	M-DMP	MIU	n/a	JSWA5	

NOTE: Furthermore, clock recovery based on Target Transmission Timestamps (e.g. RTP Time stamps for PS/TS encapsulation) is not possible when transmission rate adaptation is performed.

7.5.4.4.6.2.80.4

[GUIDELINE] If the Receiving Endpoint specifies a Target Buffer Duration using the Buffer-Info.dlna.org header (guideline 7.5.4.4.6.2.17) and the <bin> field of the a=trans-rate-adapt.dlna.org attribute is equal to 1, then Receiving Endpoint shall not attempt to recover the Serving Endpoint's clock from the RTP time stamp until the amount of data (in NPT) specified by the Target Buffer Duration has been received.

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	MIU	n/a	YZLTO	

NOTE: Example: If the SETUP request included "TD=5000" on the Buffer-Info.dlna.org header, then the Receiving Endpoint must not attempt to recover the clock until it has received the first 5 seconds worth of data in NPT time.

The NPT time can be derived from the decode time of the RTP payload.

Receiving Endpoint can still recover the Serving Endpoints clock during this time period from the Wall Clock Time in the RTP packet (if available).

7.5.4.4.6.2.81 MT RTP DLNAQOS RTSP traffic

[GUIDELINE] If DLNAQOS as defined in 7.2 is implemented, RTSP traffic in both directions (from Serving Endpoint to Receiving Endpoint and vice versa) shall be tagged with DLNAQOS_2, or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3), in accordance with Table 11.

[ATTRIBUTES]

M	R	DMS DMP DMR +PU+	M-DMS M-DMP	MIU	n/a	R9YSY	

NOTE: PAUSE and TEARDOWN are important RTSP command to stop the stream of a congested network. RTSP messages use their own TCP connection, i.e. media is not transferred by the DMS on the same connection. Because of this separate connection, RTSP request messages do not have to be at the same priority that the server will use to deliver the media.

7.5.4.4.6.2.82 MT HTTP Transport Conditions for Seek and Play Speed Operations

7.5.4.4.6.2.82.1

[GUIDELINE] A UPnP AV MediaRenderer that receives a request from a UPnP AV MediaRenderer Control Point to play at a valid speed of 's' some media resource whose first field of protocolInfo is "http-get" may send HTTP requests with PlaySpeed.dlna.org, Range, and/or TimeSeekRange.dlna.org headers against the UPnP AV MediaServer.

A valid speed 's' is one whose value is '1' or is included by the UPnP AV MediaRenderer in the play-speed-list (identified by X_DLNA_PS as defined in 7.4.1.6.29.2) of AVT.CurrentTransportActions virtual instance state variable.

[ATTRIBUTES]

O	A	DMR	n/a	n/a	[44]	2WOVM

NOTE: When a UPnP AV MediaRenderer Control Point sends the UPnP AV MediaRenderer a request to play content at some speed, the UPnP AV MediaRenderer decides the best way to satisfy the request. The UPnP AV MediaRenderer could send HTTP time/byte seek requests to the UPnP AV MediaServer, or it could perform the operation locally using cached content, or both.

7.5.4.4.6.2.82.2

[GUIDELINE] If a UPnP AV MediaRenderer indicates support for controller-time seek operations for a resource whose first field of protocolInfo is "http-get", then upon receiving an AVT:Seek request to perform such operations, the UPnP AV MediaRenderer may issue HTTP time-based seek requests against the UPnP AV MediaServer.

[ATTRIBUTES]

O	A	DMR	n/a	n/a	[44]	RQQ8Q

NOTE: This guideline clarifies that upon receiving a request from a UPnP AV MediaRenderer Control Point to perform a controller-time seek operation (on a resource associated with "http-get"), the UPnP AV MediaRenderer will decide the best way to satisfy the request. Some UPnP AV MediaRenderers might have cached the entire file and consequently, will be able to perform the request locally. Other UPnP AV MediaRenderers will send time-based seek requests to the UPnP AV MediaServer. Other UPnP AV MediaRenderers will send byte-range requests to the UPnP AV MediaServer. The UPnP AV MediaRenderer determines any necessary means to satisfy the request.

7.5.4.4.6.2.82.3

[GUIDELINE] If a UPnP AV MediaRenderer indicates support for controller-byte seek operations for a resource whose first field of protocolInfo is "http-get", then upon receiving an AVT:Seek request to perform such operations, the UPnP AV MediaRenderer may issue HTTP byte-based seek requests against the UPnP AV MediaServer.

[ATTRIBUTES]

O	A	DMR	n/a	n/a	[44]	EUSRW

NOTE: This guideline clarifies that upon receiving a request from a UPnP AV MediaRenderer Control Point to perform a controller-byte seek operation (on a resource associated with "http-get"), the UPnP AV MediaRenderer will decide the best way to satisfy the request. Some UPnP AV MediaRenderers might have cached the entire file and consequently, will be able to perform the request locally. Other UPnP AV MediaRenderers will send time-based seek requests to the UPnP AV MediaServer. Other UPnP AV MediaRenderers will send byte-range requests to the UPnP AV MediaServer. The UPnP AV MediaRenderer determines any necessary means to satisfy the request.

7.5.4.4.6.2.83 MT RTP Transport Conditions for Seek and Play Speed Operations

7.5.4.4.6.2.83.1

[GUIDELINE] A UPnP AV MediaRenderer that receives a request from a UPnP AV MediaRenderer Control Point to play at a valid speed of 's' some media resource whose first field of protocolInfo is "rtsp-rtp-udp" may send an RTSP request to the UPnP AV MediaServer.

A valid speed 's' is one whose value is '1' or is included by the UPnP AV MediaRenderer in the play-speed-list (identified by X_DLNA_PS as defined in 7.4.1.6.29.2) of AVT.CurrentTransportActions virtual instance state variable.

[ATTRIBUTES]

O	A	DMR	n/a	n/a	[27] [44]	OG3HW	

NOTE: This guideline indicates that upon receiving a request to play some resource at a speed of 's', a UPnP AV MediaRenderer could pass the request to the UPnP AV MediaServer, or it could satisfy the request using a cached copy (if available). If the UPnP AV MediaRenderer sends a request to the UPnP AV MediaServer, the UPnP AV MediaRenderer could use the RTSP Scale header, or the Speed header in RTSP Play as defined in the RTP subclause of the Guidelines.

7.5.4.4.6.2.83.2

[GUIDELINE] If a UPnP AV MediaRenderer indicates support for controller-time seek operations for a resource whose first field of protocolInfo is "rtsp-rtp-udp", then upon receiving an AVT:Seek request to perform such operations, the UPnP AV MediaRenderer may issue a request for an RTP Seek Media Operation against the UPnP AV MediaServer.

[ATTRIBUTES]

O	A	DMR	n/a	n/a	[27] [44]	WOVM6	

NOTE: This guideline clarifies that upon receiving a request from a UPnP AV MediaRenderer Control Point to perform a controller-time seek operation (on a resource associated with "rtsp-rtp-udp"), the UPnP AV MediaRenderer could issue a seek request against the server, or it could satisfy the request using a cached copy (if available).

7.6 Content Transformation Device Virtualization

7.6.1 Theory of Operations

Media requirements are different between home networked devices and mobile or handheld devices. Typically, the mobile devices have smaller screens, lower capability processors, less storage, and lower communications bandwidth. Therefore, media profiles which are tailored to this environment are especially important to mobile devices. Media servers in the home might not have content in profiles that are optimized for the handheld, and renderers within the digital home might not accept the handheld optimized profiles. In addition, media servers might not be able to interpret bitstreams of content formats that are important to the handheld devices, and hence, cannot stream that content. In order to increase the interoperability between mobile and home devices, content transformation is an important consideration of the interface between handheld devices and the digital home. Content transformation can include transcoding, transrating, or scaling of a content binary. There should exist within the network, a device that accepts media from the handheld device and makes it available in common home networking profiles, and it shall accept media from the home network and make it available to the handheld device in a profile appropriate to that environment. Within the DLNA guidelines, media is made available through the use of a Digital Media Server and consumed through Digital Media Player or Digital Media Renderer devices. Content transformation can be accomplished by making use of these same concepts to make media available in and consume media in alternate profiles. We define a "virtual" device as one that encapsulates and extends the capabilities of another device on the network. The existing device on the network is known as the "native" device. A virtual device can extend the functionality of a native device without any special relationship with that device, it uses the existing public interface to control the device when necessary. Other devices on the network can connect to the virtual device as if it were the native device, and the virtual device will control the native device to create any intermediate results or operations necessary. A virtual renderer does not have the capability to display the media itself but it encapsulates and extends a native renderer. For example, if there is a renderer available that can accept MPEG2 content (such as an HND television) a virtual media renderer can be created that can accept MPEG4 content and displays it on the television. When a control point sends an MPEG4 URL to the virtual renderer, the media is transcoded from MPEG4 to

MPEG2 and the virtual renderer uses a control point to drive the real renderer to actually display the transcoded MPEG2 content.

A similar action can be performed for media servers. If content is available in on a home DMS, a virtual server can be created which uses knowledge of content transformation that it can perform to make available that same content in alternate DLNA media format profiles. For example, when a CDS:Browse request comes in to the virtual server, it will use a control point to perform a CDS:Browse on the same container of the native server. Once it receives the metadata of the content on the native server, it can add <res> fields to the metadata representing content transformations that it can perform. The URLs of these <res> fields can point to different systems than the native server.

To allow a control point to determine if it is communicating with a virtual device, the virtual device shall specify the DLNAVIRT tag in its device description document and specify the native device that it is a virtual copy of. This will allow the control point to sort out the relationships between virtual and native devices on the network. A control point will then see a number of servers and renderers on the network with different capabilities. The control point should be aware of the concept of virtual devices and should never present to the user an interface to both the virtual device and the underlying real device. The virtual device should expose all of the capabilities of the underlying devices and make it unnecessary for the user to access the underlying device.

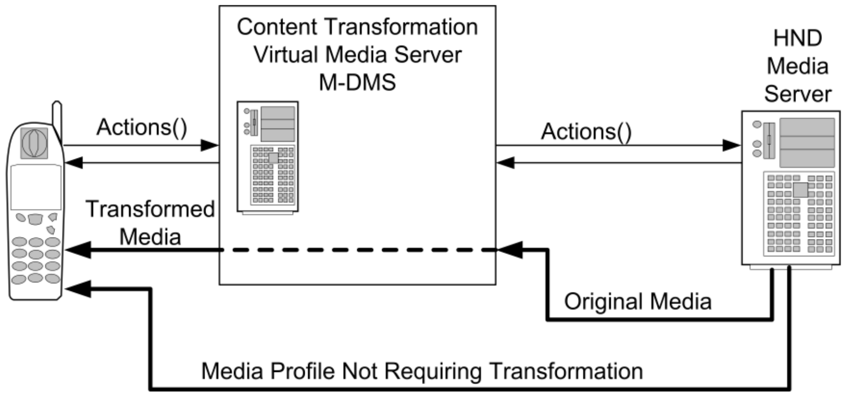


Figure 39 — Content Transformation with a Virtual MediaServer

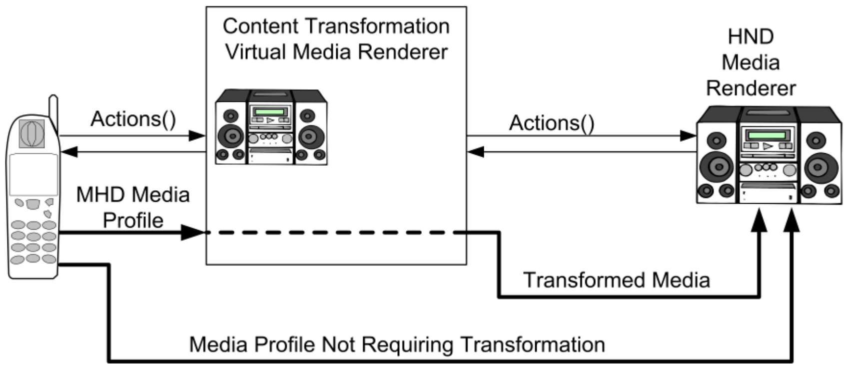


Figure 40 — Content Transformation with a Virtual MediaRenderer

7.6.2 Virtual Device Implementation

7.6.2.1 General

Any device can choose to implement or not implement virtual device functionality. This initial set of guidelines define that for any implementation that supports virtual instances, shall fully support the guidelines within this subclause

7.6.2.2 Virtual Device Conformance to Guidelines

7.6.2.2.1

[Guideline] A device may optionally implement virtual server or renderer functionality.

[Attributes]

~~8~~	~~9~~	~~10~~ DMS DMR	~~11~~ M-DMS	~~12~~ n/a	~~13~~ n/a	~~14~~ ZGPK9	

Formatted: None, No bullets or numbering, Don't keep with next, Hyphenate

NOTE: A virtual server or renderer is one which encapsulates the functionality of another device. A virtual server does not manage its own container hierarchy but relies on an underlying native server. A virtual renderer does not have direct rendering capabilities but relies on another device in the network to render content.

~~14.1.1.1.1~~7.6.2.2.2

[Guideline] If a device implements virtual server or renderer functionality, it shall adhere to all guidelines in this subclause for the appropriate virtual device.

[Attributes]

M	A	DMS DMR	M-DMS	n/a	n/a	R4XV7	

~~14.1.1.1.2~~7.6.2.2.3

[Guideline] A virtual DMS server shall adhere to all mandatory and conditionally mandatory guidelines for a DMS in addition to the guidelines contained in this subclause that are for virtual servers.

[Attributes]

M	A	DMS	n/a	n/a	n/a	8V7VH	

~~14.1.1.1.3~~7.6.2.2.4

[Guideline] A virtual M-DMS server shall adhere to all mandatory guidelines for an M-DMS in addition to the guidelines contained in this subclause that are for virtual servers.

[Attributes]

M	A	n/a	M-DMS	n/a	n/a	X34I7	

~~14.1.1.1.4~~7.6.2.2.5

[Guideline] A virtual DMR shall adhere to all mandatory guidelines for a DMR in addition to the guidelines contained in this subclause that are for virtual renderers.

[Attributes]

M	A	DMR	n/a	n/a	n/a	HZG5L	

~~14.1.2~~7.6.3 Virtual Device, Device Discovery and Control (DDC)

~~14.1.2.1~~7.6.3.1 General

A virtualized device is any UPnP device that extends and encapsulates another device. For example a virtual server may extend an existing, native, server in the network by offering additional content. A virtual renderer may extend a native renderer by offering additional input protocols and formats that are transformed on the fly to a format that the native renderer can use. This subclause of the guidelines defines how virtual devices respond to device description actions.

~~14.1.2.2~~7.6.3.2 DDC UPnP Device Description of Virtualized Device

~~14.1.2.2.1~~7.6.3.2.1

[GUIDELINE] A virtual device shall define the device(s) that it is virtualizing through the use of the <dlna:X_DLNAVIRT> XML element inside the <device> element of the device description document. The value of this element is a UUID of the original device that is being virtualized, or it is the value "*"

An example of <dlna:X_DLNAVIRT> element is shown as follows:

- <dlna:X_DLNAVIRT xmlns:dlna="urn:schemas-dlna-org:device-1-0">
  14EF6B21-7130-4525-B8C8-93FBFCF8C1A8
  </dlna:X_DLNAVIRT>

The format of a UUID is as specified in 7.3.2.19.

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[47]	86LYX

NOTE: This tag allows a control point to determine that this is a virtual device and specifies the native device that it is operating on.

~~14.1.2.2.2~~7.6.3.2.2

[GUIDELINE] The urn:schemas-dlna-org:device-1-0 namespace shall be specified for the <dlna:X_DLNAVIRT> element and the namespace prefix shall be "dlna:"

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[47]	6LYXI

~~14.1.2.2.3~~7.6.3.2.3

[GUIDELINE] The namespace prefix declaration for the dlna: namespace may be specified in the <root> element of the device description.

[ATTRIBUTES]

O	A	DMS DMR	M-DMS	n/a	[47]	34I77

~~14.1.2.2.4~~7.6.3.2.4

[GUIDELINE] The value of the UUID in the <dlna:X_DLNAVIRT> element shall match the UUID of the device that is being virtualized. This value is as specified in that device's SSDP advertisement message.

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[47]	ZG5L7

~~14.1.2.2.5~~7.6.3.2.5

[Guideline] The value of "*" in the <dlna:X_DLNAVIRT> XML element represents "all servers currently on the network".

Note that this is a dynamic set and if a native server leaves the network, the aggregate virtual server does not have to leave the network as well.

[Attributes]

M	A	DMS	M-DMS	n/a	[47]	288O9	

NOTE: This will allow aggregation -it is possible to create one virtual server which represents all content currently available on the network.

~~14.1.2.2.6~~7.6.3.2.6

[Guideline] The value of "*" in the <dlna:X_DLNAVIRT> XML element shall not be used for renderers.

[Attributes]

M	A	DMR	n/a	n/a	[47]	3RHST	

NOTE: There is no way to virtualize multiple renderers because there is no way to choose where the content is to be actually played.

~~14.1.2.2.7~~7.6.3.2.7

[Guideline] A virtual device that represents a single native device should have a device name that contains the native device's name and informs the user that this is a virtual device based upon the given native device.

[Attributes]

S	A	DMS DMR	M-DMS	n/a	[47]	F288O	

NOTE: If the DLNAVIRT tag is not understood, the device name can direct the user to realize that this is a virtual device. For example, if the name of the native server is "My_Media_Server" the virtual device could have a name such as "Mobile ready My_Media_Server". This guideline does not specify how the virtual server transforms the device name.

If this is an aggregating virtual server, it could have a name such as "Mobile Media Server"

Since this is only given to the user and is not interpreted by software, having various mechanisms does not cause an interoperability issue.

~~14.1.2.2.8~~7.6.3.2.8

[Guideline] A virtual server that aggregates content from multiple native servers shall have a unique name on the network that represents its function as an aggregating virtual server performing content transformation.

[Attributes]

M	A	DMS	M-DMS	n/a	[47]	P8V7V	

~~14.1.2.2.9~~7.6.3.2.9

[Guideline] The virtual device's name shall allow localized native device names to be included as part of the text of the virtual device's name.

This does not require the virtual device's name to match the language of the native device's name, only that the portion of the native device's name that is included in the virtual device's name, shall be able to be in a localized language, and that language shall be preserved in the portion of the copied name.

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[47]	E86LY	

~~14.1.2.3~~7.6.3.3 DDC UPnP Actions

~~14.1.2.3.1~~7.6.3.3.1

[GUIDELINE] The virtual device shall receive actions and relay them to the native device by use of a control point implemented in the device hosting the virtual server or renderer.

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[47]	MX34I	

NOTE: If the native device is a 1.0 device, the control point must adhere to 1.0 calling conventions and requirements, if a 1.5 device, it must adhere to that spec. etc. The requirements are set by the underlying native device.

~~14.1.2.3.2~~7.6.3.3.2

[GUIDELINE] When relaying an action, the control point implemented in the device hosting the virtual server or renderer shall adhere to all guidelines for the version and types of the device that it is calling.

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[47]	UE86L	

~~14.1.2.4~~7.6.3.4 DDC UPnP Device Description ssdp:byebye of Virtual Device

~~14.1.2.4.1~~7.6.3.4.1

[GUIDELINE] A virtual server or renderer bound to a single native device shall issue its own ssdp:byebye message within 5 seconds of receiving the native device's ssdp:byebye.

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[47]	ZP8V7	

~~14.1.2.4.2~~7.6.3.4.2

[GUIDELINE] A virtual server or renderer bound to a single native device shall issue a ssdp:byebye if it fails to receive advertisements from the native device within a CACHE-CONTROL interval. It shall issue this ssdp:byebye within 5 seconds of the end of the CACHE-CONTROL interval .

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[47]	2ZP8V	

~~14.1.2.4.3~~7.6.3.4.3

[GUIDELINE] A virtual server that aggregates content from multiple native servers shall issue its own ssdp:byebye message within 5 seconds of the last native device that it is virtualizing issuing a ssdp:byebye.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[47]	V7VHR

~~14.1.2.4.4~~7.6.3.4.4

[GUIDELINE] A virtual server that aggregates content from multiple native servers shall issue its own ssdp:byebye message within 5 seconds of recognizing that all native servers' CACHE-CONTROL intervals have expired without receiving an *advertisement set*.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[47]	LYXIA

~~14.1.2.4.5~~7.6.3.4.5

[GUIDELINE] A virtual device which has any pending UPnP requests at the time that the virtual device receives the ssdp:byebye from the native device should respond to the UPnP requests with an error 503 (Service Unavailable).

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[47]	4I77T

~~14.1.2.5~~7.6.3.5 DDC Virtual Devices

[GUIDELINE] An endpoint shall never create a virtual server or renderer for a device that is itself a virtual device.

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	n/a	I77T3

NOTE: This could create a loop in the graph of network devices.

~~14.1.3~~7.6.4 Virtual Device Media Management (MM)

~~14.1.3.1~~7.6.4.1 General

This subclause of the guidelines defines how virtual servers and renderers interact at the media management layer.

~~14.1.3.2~~7.6.4.2 CMS Action Requirement for Virtual Devices

~~14.1.3.2.1~~7.6.4.2.1

[GUIDELINE] A virtual device shall define the input media format profiles that it can accept through the use of the CMS:X_GetDLNAInputProfiles action.

The action's definition in the service description is defined below.

- <action>
- <name>X_GetDLNAInputProfiles</name>
- <argumentList>
- <argument>
- <name>InputProfiles</name>
- <direction>in</direction>
- <relatedStateVariable>X_A_ARG_Type_InputProfiles</relatedStateVariable>

- </argument>
- <argument>
- <name>SupportedInputProfiles</name>
- <direction>out</direction>
- <relatedStateVariable>X_A_ARG_Type_SupportedInputProfiles</relatedStateVariable>
- </argument>
- </argumentList>
- </action>

The X_A_ARG_TYPE_InputProfiles and X_A_ARG_TYPE_SupportedInputProfiles state variables are defined below.

- <stateVariable sendEvents="no">
- <name>X_A_ARG_Type_InputProfiles</name>
- <dataType>string</dataType>
- </stateVariable>
- <stateVariable sendEvents="no">
- <name>X_A_ARG_Type_SupportedInputProfiles</name>
- <dataType>string</dataType>
- </stateVariable>

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[45]	G5L7T

NOTE: Note: The use of the CMS:X_GetDLNAInputProfiles or CMS:X_GetDLNAOutputProfiles actions do not guarantee that all content on a native server can be made available in all of the media profiles listed, nor that a virtual renderer can accept any of the listed input profiles for transformation to any native renderer in the network.

These guidelines are intended to allow control points to quickly find virtual servers and renderers that might have content optimized for the device at the time that the UPnP device discovery occurs. It is not intended to imply that all content will be available in these formats from a virtual server or that the virtual renderer can accept these formats for all native renderers in the network.

It is intended as a general description of the types of media that a control point can expect to find on this server. This is useful when a control point attempts to locate a virtual server with a particular type of specialized media format profiles, which it will then explore in more detail for the supported media formats for each content binary.

The *InputProfiles* input argument is an unordered, comma separated list of DLNA media format profile names.

The *SupportedInputProfiles* output argument is an unordered, comma separated list of DLNA media format profile names.

- If the *InputProfiles* input argument is not empty, then SupportedInputProfiles contains all DLNA media format profiles that this virtual server can support as input for transformation that are also listed in the *InputProfiles* input argument.
- Or, in case of an empty InputProfiles value, the SupportedInputProfiles list must contain the complete list of DLNA media format profiles that this virtual device can accept as input for transformation.

For a virtual renderer SupportedInputProfiles will be the profiles that it can accept from a control point for transforming. For a virtual server, SupportedInputProfiles will be the profiles that can be read from a native server for transformation.

The response behavior is summarized in the following way.

If InputProfiles is empty, then SupportedInputProfiles contains a complete list of profiles that the virtual device is able to transform. Control points specify an empty value for InputProfiles when they want to acquire the full profile set.

If InputProfiles contains one or more profiles, then SupportedInputProfiles contains the subset of InputProfiles that the virtual device is able to transform. Control points specify one or more profiles for InputProfiles when they are interested finding out if the virtual device is able to transform certain profiles.

~~14.1.3.2.2~~7.6.4.2.2

[GUIDELINE] A virtual device shall define the output media format profiles that it supports through the use of the CMS:X_GetDLNAOutputProfiles action.

The action's definition in the service description is :

```
<action>
   <name>X_GetDLNAOutputProfiles</name>
   <argumentList>
      <argument>
         <name>OutputProfiles</name>
         <direction>in</direction
         <relatedStateVariable>
            X_A_ARG_Type_OutputProfiles
         </relatedStateVariable>
      </argument>
      <argument>
         <name>SupportedOutputProfiles</name>
         <direction>out</direction>
         <relatedStateVariable>
            X_A_ARG_Type_SupportedOutputProfiles
         </relatedStateVariable>
      </argument>
   </argumentList>
</action>
```

The X_A_ARG_TYPE_OutputProfiles and X_A_ARG_Type_SupportedOutputProfiles state variables are defined:

```
<stateVariable sendEvents="no">
   <name>X_A_ARG_Type_OutputProfiles</name>
   <dataType>string</dataType>
</stateVariable>
<stateVariable sendEvents="no">
   <name>X_A_ARG_Type_SupportedOutputProfiles </name>
   <dataType>string</dataType>
</stateVariable>
```

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[45]	L7TSG	

NOTE: The OutputProfiles intput argument is an unordered, comma separated list of DLNA media format profile names.

The *SupportedOutputProfiles* output argument is an unordered, comma separated list of DLNA media format profile names.

- If the *OutputProfiles* input argument is not empty, then SupportedOutputProfiles contains all DLNA media format profiles that this virtual server can support as output from transformation that are also listed in the *OutputProfiles* input argument.
- Or, in case of an empty OutputProfiles value, the SupportedOutputProfiles list must contain the complete list of DLNA media format profiles that this virtual device can support as output from transformation.

For a virtual renderer SupportedOutputProfiles will be the profiles that it can transform content to for output to a native renderer. For a virtual server, SupportedOutputProfiles will be the profiles that can be made available as alternate media format profiles.

The response behavior is summarized in the following way.

If OutputProfiles is empty, then SupportedOutputProfiles contains a complete list of profiles that the virtual device is able to create during a transformation from one or more profiles. Control points specify an empty value for OutputProfiles when they want to acquire the full profile set.

If OutputProfiles contains one or more profiles, then SupportedOutputProfiles contains the subset of OutputProfiles that the virtual device is able to create during a transformation from one or more profiles. Control points specify one or more profiles for OutputProfiles when they are interested finding out if the virtual device creates those profiles.

~~14.1.3.2.3~~7.6.4.2.3

[GUIDELINE] A virtual device may optionally define the content transformations that it can perform through the use of the CMS:X_GetDLNATransformProfiles action.

[ATTRIBUTES]

O	A	DMS DMR	M-DMS	n/a	n/a	88O9X

~~14.1.3.2.4~~7.6.4.2.4

[GUIDELINE] The CMS:X_ GetDLNATransformProfiles action's definition in the service description shall be as follows:

```
<action>
   <name>X_GetDLNATransformProfiles</name>
   <argumentList>
      <argument>
         <name>TransformProfiles</name>
         <direction>in</direction>
         <relatedStateVariable>
            X_A_ARG_Type_TransformProfiles
         </relatedStateVariable>
      </argument>
      <argument>
         <name>SupportedTransformProfiles</name>
         <direction>out</direction>
         <relatedStateVariable>
            X_A_ARG_Type_SupportedTransformProfiles
         </relatedStateVariable>
      </argument>
   </argumentList>
</action>
```

The X_A_ARG_TYPE_TransformProfiles and X_A_ARG_TYPE_SupportedTransformProfiles state variables are defined as follows:

```
<stateVariable sendEvents="no">
   <name>X_A_ARG_Type_TransformProfiles</name>
   <dataType>string</dataType>
</stateVariable>
<stateVariable sendEvents="no">
   <name>X_A_ARG_Type_SupportedTransformProfiles</name>
   <dataType>string</dataType>
</stateVariable>
```

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[45]	5L7TS

NOTE: The *TransformProfiles* input argument is an unordered, comma separated list of ordered pairs of DLNA media format profile names.

The *SupportedTransformProfiles* output argument is an unordered, comma separated list of ordered pairs of DLNA media format profile names.

The SupportedTransformProfiles list contains the ordered pairs of DLNA media format profile names that is described by this boolean statement of (([a] AND [b]) OR ([c])):

- If the ordered pairs listed in the *TransformProfiles* input argument is not empty, then SupportedTransformProfiles contains all ordered pairs that this virtual server can support as transformations that are also listed in the *TransformProfiles* input argument.
- Or, in case of an empty TransformProfiles value, the SupportedTransformProfiles list must contain the complete list of ordered pairs that this virtual device can support for transformations.

An ordered pair is a pair of DLNA media format profile names such that the first profile (i.e. transform-from) can be transformed into the second media format profile (i.e. transform-to). Formally, it is defined with this syntax:

- order-pair = transform-from ":" transform-to
- transform-from = pn-value
- transform-to = pn-value
- pn-value = <syntax defined in 7.4.1.3.17 MM pn-param (DLNA.ORG_PN Parameter)>

The response behavior is summarized in the following way.

If TransformProfiles is empty, then SupportedTransformProfiles contains a complete list of ordered pairs that the virtual device is able to transform. Control points specify an empty value for TransformProfiles when they want to acquire the full set of possible transforms.

If TransformProfiles contains one or more ordered pairs, then SupportedTransformProfiles contains the subset of TransformProfiles that the virtual device is able to transform. Control points specify one or more ordered pairs for TransformProfiles when they are interested finding out if the virtual device supports a particular set of transforms.

~~14.1.3.3~~7.6.4.3 MM Virtual Server

~~14.1.3.3.1~~7.6.4.3.1

[GUIDELINE] If a virtual server aggregates content from multiple native servers, it shall aggregate content from all native servers currently on the network and shall specify the "*" flag in the <dlna:X_DLNAVIRT> XML element of its device description.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	8O9XK

~~14.1.3.3.2~~7.6.4.3.2

[GUIDELINE] All virtual servers shall support the required UPnP components of a DMS or M-DMS, including all required actions and state variables.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	STLPO

NOTE: Specifically, it must adhere to the following guideline requirements for components.

- 7.4.1.2.11 MM DMS/M-DMS UPnP AV MediaServer Device Definition
- 7.4.1.2.12 MM DMS/M-DMS ContentDirectory Rules
- 7.4.1.2.13 MM DMS/M-DMS ConnectionManager Rules

~~14.1.3.3.3~~7.6.4.3.3

[GUIDELINE] A virtual server that does not aggregate content from multiple native servers shall support all actions that the underlying native server supports.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	O9XKH

NOTE: The reason for this is that we want to avoid the situation where the control point goes to the virtual server and finds that it can't perform a critical action. It then must locate the native server and perform that action on the native server. It is a better solution for the control point to be assured that it can perform all actions by just working with the virtual server.

~~14.1.3.3.4~~7.6.4.3.4

[GUIDELINE] A virtual server that does not aggreage content from multiple native servers shall make available all of the events of the native server.

The control point on the virtual server shall subscribe for events on the native server, and when the event occurs on the native server, it shall be forwarded as if it had occurred on the virtual server.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	HSTLP	

~~14.1.3.3.5~~7.6.4.3.5

[GUIDELINE] A virtual server that aggregates content from multiple native servers may limit the actions and events that it supports.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	9OV47	

NOTE: If some of the native servers do not support optional actions - such as Search it is impossible for the virtual server to supply the necessary functionality.

~~14.1.3.3.6~~7.6.4.3.6

[GUIDELINE] A UPnP action on a virtual server shall fail if it cannot meet the timing restrictions for UPnP actions even if the underlying UPnP action on the native server succeeds. See guideline 7.3.2.9 for UPnP device responsiveness.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	TLPOH	

~~14.1.3.3.7~~7.6.4.3.7

[GUIDELINE] A virtual server shall return a result that is within the size limit of UPnP results. See guideline 7.3.2.17 for UPnP SOAP packet size.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	7SMU8	

NOTE: The virtual server is free to truncate the response from the native server at an appropriate boundary so that the final return value from the virtual server will fit within the space constraints.

~~14.1.3.3.8~~7.6.4.3.8

[GUIDELINE] The CMS.SourceProtocolInfo variable of a virtual server shall comprise the protocolInfos listed in the CMS.SourceProtocolInfo of the native server(s) and protocolInfos corresponding to the profiles listed in the *SupportedOutputProfiles* output argument of the virtual server's CMS: X_GetDLNAOutputProfiles when the OutputProfiles argument is set to an empty string.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	RHSTL	

NOTE: See 7.6.4.2.2 (CMS Action Requirement for Virtual Devices) for the more information about CMS:X_GetDLNAOutputProfiles.

~~14.1.3.3.9~~7.6.4.3.9

[GUIDELINE] If the virtual server supports optional content management (OCM) operations - see guideline 7.4.1.7.2, it shall control the native server as a valid UPnP AV MediaServer Control point for the given operation. Specifically, it shall adhere to the control point portions of the following guidelines.

- 7.4.1.7.11 MM/CM: Upload AnyContainer Operation
- 7.4.1.7.12 MM/CM OCM: Upload Content Operation
- 7.4.1.7.13 MM/CM: OCM: Create Child Container Operation
- 7.4.1.7.14 MM/CM: OCM: Destroy Object Operation
- 7.4.1.7.15 MM/CM: Use of Valid Values
- 7.4.1.7.19 MM/CM: General Rule for Creating <res> Elements: Content Transfer Process
- 7.4.1.7.20 MM/CM: General Rule for Creating <res> Elements: Resume Content Transfer Process
- 7.4.1.7.23 MM/CM: General Rules for CDS:CreateObject Request Syntax
- 7.4.1.7.26 MM/CM: Content Transfer Process
- 7.4.1.7.28 MM/CM: Auto-Destroy Behavior for a Failed or Partial Content Transfer Process

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	47SMU

~~14.1.3.3.10~~7.6.4.3.10

[GUIDELINE] Any action on a virtual server shall fail if the corresponding operation on the native server fails and the virtual server shall return the same error message as the native server.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	WA5NX

NOTE: An operation in this context might consist of a number of UPnP actions - for example an implementation of an action on a virtual server is free to call multiple actions on the native server. Some of these actions on the native server might fail. However, taken as a whole, all of the calls to the native server represent a single operation.

For example, the virtual server can make several calls to the native server to test the level of support for media profiles. Some of these calls might fail if the native server does not support a format, but overall, the operation will succeed when a compatible format is found. .

~~14.1.3.3.11~~7.6.4.3.11

[GUIDELINE] Any action on a virtual server will be declared successful only if the native server operation succeeds.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	V47SM

~~14.1.3.3.12~~7.6.4.3.12

[GUIDELINE] If an operation occurs on the DLNA.ORG_AnyContainer on the virtual server, the virtual server shall map that to a corresponding operation on the DLNA.ORG_AnyContainer of one of the native servers that it is virtualizing.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	NXTOX	

NOTE: If the virtual server is aggregating multiple native servers, it must choose one to apply the operation to. If it is not aggregating, there is a 1-to-1 mapping to the native server.

~~14.1.3.3.13~~7.6.4.3.13

[GUIDELINE] If the native server can accept the incoming media format profile of an upload operation, then the ImportURI returned by the virtual server shall point to the native server and the upload content transfer process shall occur directly to the native server.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	OV47S	

~~14.1.3.3.14~~7.6.4.3.14

[GUIDELINE] If the native server cannot accept the incoming media format profile of an upload operation, then the Import URI shall point to the virtual server. It shall accept the content through a content transfer process and transform it to a format that the native server can support, and place the transformed content on the native server.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	SWA5N	

~~14.1.3.3.15~~7.6.4.3.15

[GUIDELINE] The upload of content to the virtual server shall fail if the virtual server cannot upload transformed content to the native server or the native server cannot accept the transformed content.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	A5NXT	

~~14.1.3.3.16~~7.6.4.3.16

[GUIDELINE] If a virtual server receives content that it cannot place on the native server it shall fail the Media Transport operation with the same error response as returned from the native server.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	T6L4F	

~~14.1.3.3.17~~7.6.4.3.17

[GUIDELINE] If a virtual server receives content via an HTTP POST operation, the virtual server shall delay the final response on the final chunk of received data until the media is (possibly transformed) and stored on the native server

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	5NXTO

NOTE: This ensures that all incoming media is correctly received and accepted by the native server before sending a final acceptance.

~~14.1.3.3.18~~7.6.4.3.18

[GUIDELINE] If a virtual server is aggregating content from a number of native servers, and one of the native servers leaves the network, any query issued more than 1 second after the native server leaves the network shall not show that content in the hierarchy.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	6L4FE

NOTE: A native server can leave the network by issuing a ssdp:byebye message or CACHE-CONTROL seconds can elapse without seeing an *advertisement set*.

~~14.1.3.3.19~~7.6.4.3.19

[GUIDELINE] For every CDS object on the native server, the virtual server shall advertise that content with the original content format, scaling, and rate available through a direct URL reference to the native server.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	LTOT6

NOTE: This is required so that a user doesn't have to manipulate one profile of content on one server and switch to a different server to manipulate a different format of the same content. A v1.0 DMP will only be able to use direct URL references.

~~14.1.3.3.20~~7.6.4.3.20

[GUIDELINE] For every CDS object on the native server, the virtual server may advertise that content with the original content format, scaling, and rate available through an indirect URL reference to the native server, using the PlaySingleURI mechanism as specified in 7.4.1.4.26 MM CDS DLNA PlaySingle URI Values.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	B73N2

~~14.1.3.3.21~~7.6.4.3.21

[GUIDELINE] A virtual server shall use additional <res> elements on a CDS object to advertise alternate profiles or alternate data rates, or alternate media scalings.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	OT6L4	

NOTE: The virtual server must not create new content entries to represent the transcoded content.

~~14.1.3.3.22~~7.6.4.3.22

[GUIDELINE] New <res> elements shall not be advertised until the virtual server can correctly respond to a request for the content in the indicated media parameters.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	TOT6L	

NOTE: There are no static reason why the content cannot be served when requested. For instance, if offline content transformation is performed and the transformed file is not available, a <res> element would not be published.

Don't use dynamic conditions, such as network bandwidth, processor resources, etc. that can change rapidly, to determine whether a <res> field is provided or not.

~~14.1.3.3.23~~7.6.4.3.23

[GUIDELINE] If offline transformation of content is performed, the <res> element shall not be published in the response to a CDS:Browse or CDS:Search until the transformation is complete and the content binary is available.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	DUTU4	

~~14.1.3.3.24~~7.6.4.3.24

[GUIDELINE] If real time conversion of content is performed, the <res> element shall not be published in the response to a CDS:Browse or CDS:Search until the transformation subsystem is ready to respond to requests for content binaries.

The request for content binaries may fail due to dynamic reasons; however the transformation service shall be ready to respond with an appropriate failure condition.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	3V6TC	

~~14.1.3.3.25~~7.6.4.3.25

[GUIDELINE] If a real time transformation cannot be completed when requested due to dynamic conditions on the virtual server, the media transport layer shall issue an appropriate error message within the transport protocol requested

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	W3V6T	

~~14.1.3.3.26~~7.6.4.3.26

[GUIDELINE] New <res> elements shall advertise URI values that allow for the virtual server to setup the requested content transformation.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	GB73N	

NOTE: If a realtime transform is performed, information in the URI can be used to define the server and URI of the original content.

~~14.1.3.3.27~~7.6.4.3.27

[GUIDELINE] If a virtual server cannot create a URI value for content that meets the above guideline and also meets the maximum allowable URI size restriction, it shall not publish a <res> element for this content transformation

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	TGB73	

~~14.1.3.3.28~~7.6.4.3.28

[GUIDELINE] A virtual server shall retain all recommended metadata (as specified by guideline 7.4.1.3.11.3) that is available for a CDS object.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	EDUTU	

NOTE: The virtual server can choose to delete other metadata entries at its discretion.

~~14.1.3.3.29~~7.6.4.3.29

[GUIDELINE] A virtual server shall specify the available media operations in the 4th field of a protocolInfo on a <res> element to transformed content.

[ATTRIBUTES]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	VW3V6	

NOTE: For example, the native server might be able to support fast forward playback, while the transformed ontent cannot, so for the new <res> elements, they must have the correct corresponding set of media operations that the virtual server can support.

~~14.1.3.3.30~~7.6.4.3.30

[GUIDELINE] A virtual server may reduce the set of media operations in the 4th field of a protocolInfo for a <res> element added for transformed content.

[ATTRIBUTES]

O	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	6EDUT	

~~14.1.3.3.31~~7.6.4.3.31

[GUIDELINE] The virtual server shall copy the entire protocolInfo unaltered for a <res> element where the URI is a direct reference to the native server's content.

[Attributes]

M	A	DMS	M-DMS	n/a	[44] [45] [46] [49]	96EDU

NOTE: The URI still points to the native server. The availability of that server's media operations is independent of the source of the content directory.

~~14.1.3.4~~7.6.4.4 MM Virtual Renderer

~~14.1.3.4.1~~7.6.4.4.1

[Guideline] This Guideline no longer applies.

[Attributes]

n/a	n/a	n/a	n/a	n/a	n/a	6VW3V

~~14.1.3.4.2~~7.6.4.4.2

[Guideline] A virtual renderer shall support the required UPnP components of a DMR, including all required actions and state variables.

[Attributes]

M	A	DMR	n/a	n/a	[44] [45] [48] [90]	O96ED

NOTE: Specifically, it must adhere to the following guideline requirements for components.

- 7.4.1.2.5 MM DMR UPnP AV MediaRenderer Device Definition
- 7.4.1.2.7 MM DMR ConnectionManager Rules
- 7.4.1.2.8 MM DMR RenderingControl Rules

~~14.1.3.4.3~~7.6.4.4.3

[Guideline] A virtual renderer shall make available all of the events of the native renderer.

The control point on the virtual renderer shall subscribe to the events on the native and when the event occurs on the native renderer, it shall be forwarded as if it had occurred on the virtual renderer.

[Attributes]

M	A	DMR	n/a	n/a	[44] [45] [48] [90]	Y6VW3

~~14.1.3.4.4~~7.6.4.4.4

[Guideline] The CMS.SinkProtocolInfo variable of a virtual renderer shall comprise the protocolInfos listed in the CMS.SinkProtocolInfo of the native renderer and protocolInfos corresponding to the profiles listed in the *SupportedInputProfiles* output argument of the virtual renderer's CMS:X_GetDLNAInputProfiles response when the InputProfiles argument is set to an empty string.

[Attributes]

M	A	DMR	n/a	n/a	[44] [45] [48] [90]	9TGB7

NOTE: See 7.6.4.2.1 (CMS Action Requirement for Virtual Devices) for the more information about CMS:X_GetDLNAInputProfiles.

~~14.1.3.4.5~~7.6.4.4.5

[GUIDELINE] A virtual renderer shall be bound to a single real renderer.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	[44] [45] [48] [90]	K9TGB	

~~14.1.3.4.6~~7.6.4.4.6

[GUIDELINE] A virtual renderer may buffer any reasonable amount of data for a transformation before starting the playback on the native renderer.

[ATTRIBUTES]

O	A	DMR	n/a	n/a	[44] [45] [48] [90]	4BSTW	

NOTE: Due to items like network bandwidth, jitter, or capabilities of the content transformation engine, the virtualizer might need to buffer a substantial portion of the content before starting the playback.

~~14.1.4~~7.6.5 Virtual Device Media Formats (MF)

~~14.1.4.1~~7.6.5.1 MF Virtual HND Server Media Types

~~14.1.4.1.1~~7.6.5.1.1

[GUIDELINE] A virtual DMS server shall support at least one HND required media format profiles for the media classes that it supports.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[56]	C5K83	

NOTE: See 6.2 in **DLNA__Media_Formats, 6.2**.

~~14.1.4.1.2~~7.6.5.1.2

[GUIDELINE] A virtual DMS server shall make additional media format profiles available as DLNA media format profiles.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	[56]	24N2G	

~~14.1.4.2~~7.6.5.2 MF Virtual MHD Server Media Types

~~14.1.4.2.1~~7.6.5.2.1

[GUIDELINE] A virtual M-DMS server shall support at least one MHD required media format profile for the media classes that it supports.

[ATTRIBUTES]

M	A	n/a	n/a	M-DMS	[56]	EV3PV	

NOTE: See 6.2 in [56].

~~14.1.4.2.2~~7.6.5.2.2

[GUIDELINE] A virtual M-DMS server shall make additional media format profiles available as DLNA media format profiles

[ATTRIBUTES]

M	A	n/a	n/a	M-DMS	[56]	XVALG	

~~14.1.4.3~~7.6.5.3 MF Virtual HND HND Renderer Media Types

[GUIDELINE] A virtual DMR shall support all required media format profiles for the media classes that it consumes.

[ATTRIBUTES]

M	A	n/a	DMR	n/a	[56]	86X5J	

NOTE: See See 6.2 in [56].

~~14.1.5~~7.6.6 Virtual Device Media Transport (MT)

~~14.1.5.1~~7.6.6.1 MT Virtual HND Server Media Transport

[GUIDELINE] A virtual DMS server shall support transport of the media over all HND required media transport protocols.

[ATTRIBUTES]

M	A	DMS	n/a	n/a	n/a	HR6EO	

NOTE: This reiterates the requirement that a virtual DMS must implement all of the mandatory requirements for a native DMS.

~~14.1.5.2~~7.6.6.2 MT Virtual MHD Server Media Transport

[GUIDELINE] A virtual M-DMS server shall support transport of the media over all required MHD media transport protocols.

[ATTRIBUTES]

M	A	n/a	M-DMS	n/a	n/a	UTU4B	

NOTE: This reiterates the requirement that a virtual M-DMS must implement all of the mandatory requirements for a native M-DMS.

~~14.1.5.3~~7.6.6.3 MT Virtual HND HND Renderer Media Types

[GUIDELINE] A virtual DMR shall accept content over all required media transport protocols.

[ATTRIBUTES]

M	A	DMR	n/a	n/a	n/a	6TC5K	

NOTE: This reiterates the requirement that a virtual DMR must implement all of the mandatory requirements for a native DMR.

~~14.1.5.4~~7.6.6.4 MT Virtual Device Control

[GUIDELINE] A virtual device shall control the native device with the version of HTTP that is supported by the native device (1.0 or 1.5).

[ATTRIBUTES]

M	A	DMS DMR	M-DMS	n/a	[21] [23] [33]	N24N2	

NOTE: This reiterates the requirement that a virtual device must interact with the native device using the appropriate HTTP version.

~~14.2~~7.7 Media Interoperability Unit (MIU)

~~14.2.1~~7.7.1 General

This set of guidelines builds upon the concept of virtual servers and renderers in order to support media interoperability between mobile handheld devices and DMS (Digital Media Server) and DMR (Digital Media Renderer) devices in the HND Device Category. The definition of the DMS and DMR define media requirements that might not be optimal for a mobile handheld device (LPCM audio and MPEG2 video). The MIU ensures interoperability between mobile handheld devices and is responsible for the media transforms that bridge the gap between the MHD required formats and the HND required formats. The definition of a virtual server and virtual renderer are in subclause 7.6.

~~14.2.2~~7.7.2 Media Interoperability Unit Media Management Guidelines

~~14.2.2.1~~7.7.2.1 MM Media Interoperability Using Virtual Devices

~~14.2.2.1.1~~7.7.2.1.1

[GUIDELINE] An MIU shall expose all content available on all DMS and M-DMS devices currently available on the network by making available a virtual M-DMS for each native DMS currently available on the network and making available a virtual DMS for each native M-DMS currently available on the network.

[ATTRIBUTES]

M	A	n/a	n/a	MIU	n/a	U4BST	

~~14.2.2.1.2~~7.7.2.1.2

[GUIDELINE] The MIU may optionally make available an aggregating virtual server that exposes content from all native DMS and M-DMS devices curreltly available on the network.

[ATTRIBUTES]

O	A	n/a	n/a	MIU	n/a	TU4BS	

~~14.2.2.1.3~~7.7.2.1.3

[GUIDELINE] Both Audio and AV media classes should be supported by an MIU virtual server.

[ATTRIBUTES]

S	A	n/a	n/a	MIU	n/a	TC5K8	

~~14.2.2.1.4~~7.7.2.1.4

[GUIDELINE] For each native DMR currently available on the network, an MIU shall make available a virtual DMR that accepts the MHD mandatory media profiles of the DLNA media classes supported by the native DMR.

[ATTRIBUTES]

M	A	n/a	n/a	MIU	n/a	V6TC5	

~~14.2.2.1.5~~7.7.2.1.5

[GUIDELINE] The MIU should reside on HND connectivity domain.

[ATTRIBUTES]

S	A	n/a	n/a	MIU	n/a	3N24N	

~~14.2.2.1.6~~7.7.2.1.6

[GUIDELINE] The MIU shall never create a virtual server or renderer for a device which is itself a virtualization.

[ATTRIBUTES]

M	A	n/a	n/a	MIU	n/a	73N24	

~~14.2.2.1.7~~7.7.2.1.7

[GUIDELINE] If an additional DMS, M-DMS, or DMR enters the network that requires content transformation, the MIU shall generate the UPnP *advertisement set* (as defined in guideline 7.3.2.4 DDC UPnP Discovery Robustness) for the virtual device(s) within 10 seconds.

[ATTRIBUTES]

M	A	n/a	n/a	MIU	n/a	FEV3P	

NOTE: The advertisement of the server device must be made within 10 seconds, however guideline 7.6.4.3.22 defines that <res> elements can not be made available until the system is ready to supply the indicated content.

~~14.2.2.1.8~~7.7.2.1.8

[GUIDELINE] Virtual servers and renderers created by the MIU shall adhere to all guidelines published in 7.3.2.4.

[ATTRIBUTES]

M	A	n/a	n/a	MIU	n/a	4FEV3	

~~14.2.2.2~~7.7.2.2 MT Transport Protocol Usage

[GUIDELINE] The virtual servers and renderers created by the MIU shall be able to transport media in the baseline transport protocol (HTTP).

[ATTRIBUTES]

M	A	n/a	n/a	MIU	n/a	L4FEV	

~~14.2.2.3~~7.7.2.3 MF Audio Content Transformation - Server

~~14.2.2.3.1~~7.7.2.3.1

[GUIDELINE] If the MIU supports the audio media class and content is available on a DMS in the HND mandatory audio media format profile, the virtual M-DMS server exposing that content on an MIU shall make it available in one of the MHD mandatory audio media format profiles.

The HND mandatory audio media format profile is as defined in 6.2.3 in See 6.2 in [56].

The MHD mandatory audio media format profiles are defined in 6.2.5 in See 6.2 in [56].

[ATTRIBUTES]

M	A	n/a	n/a	MIU	[56]	OXVAL	

~~14.2.2.3.2~~7.7.2.3.2

[GUIDELINE] If the MIU supports the audio media class and content is available on an M-DMS in any MHD mandatory audio media format profile, the virtual DMS server exposing that content on an MIU shall make it available in the HND mandatory audio media format profile.

The HND mandatory audio media format profile is as defined in 6.2.3 in See 6.2 in [56].

The MHD mandatory audio media format profiles are defined in 6.2.5 in See 6.2 in [56].

[ATTRIBUTES]

M	A	n/a	n/a	MIU	[56]	U86X5	

~~14.2.2.4~~7.7.2.4 **MF Audio Content Transformation - Renderer**

[GUIDELINE] For every DMR supporting the Audio media class, the MIU shall advertise a virtual DMR renderer that supports and accepts media in all MHD mandatory audio media format profiles

The MHD mandatory audio media format profiles are defined in 6.2.5 in [56].

[ATTRIBUTES]

M	A	n/a	n/a	MIU	[56]	TOXVA	

~~14.2.2.5~~7.7.2.5 **MFAV Content Transformations - Server**

~~14.2.2.5.1~~7.7.2.5.1

[GUIDELINE] If the MIU supports the AV media class and content is available on a DMS in the HND mandatory AV media format profile for the given region, the virtual M-DMS server exposing that content on an MIU shall make it available in the MHD mandatory AV media format profile

The HND mandatory AV media format profile is as defined in 6.2.7 in [56]. The HND AV media format profile requirements per region are defined in 6.2.7 in [56]. The MHD mandatory AV media format profile is defined in 6.2.9 in [56].

[ATTRIBUTES]

M	A	n/a	n/a	MIU	[56]	MU86X	

~~14.2.2.5.2~~7.7.2.5.2

[GUIDELINE] If the MIU supports the AV media class and content is available on an M-DMS in the MHD mandatory AV media format profile, the virtual DMS server exposing that content on the MIU shall make it available in the HND mandatory AV media format profiles for the given region.

The HND mandatory AV media format profile requirements per region and per device type are as defined in 6.2.7 in [56]. The HND AV media format profile requirements per region are defined in 6.2.7 in [56]. The MHD mandatory AV media format profile is defined in 6.2.9 in [56].

[ATTRIBUTES]

M	A	n/a	n/a	MIU	[56]	XTOXV	

~~14.2.2.6~~7.7.2.6 MF AV Content Transformation - Renderer

[GUIDELINE] For every DMR renderer supporting the AV media class, the MIU shall advertise a virtual DMR that supports and accepts media in the MHD mandatory AV media format profile.

The MHD mandatory AV media format profile is defined in 6.2.9 in [56].

[ATTRIBUTES]

M	A	n/a	n/a	MIU	[56]	SMU86

~~14.3~~7.8 Remote User Interfaces

~~14.3.1~~7.8.1 General

This subclause of the DLNA Home Networked Device Interoperability Guidelines covers the guidelines for implementing Remote User Interfaces in a DLNA compliant way.

This subclause is based on CEA-2014 revision A [101]. In this respect, it is important to note that for any reference to requirements in CEA-2014-A, a mandatory requirement in CEA-2014-A remains mandatory for DLNA, and recommended and optional requirements in CEA-2014-A remain recommended and optional in DLNA, unless explicitly specified differently in these DLNA guidelines.

~~14.3.2~~7.8.2 Remote User Interface Guidelines (RUI)

~~14.3.2.1~~7.8.2.1 Discovery of Remote UI Devices

This subclause defines the requirements for the discovery of RUI Source capabilities and RUI Sink capabilities (as defined in 5.7.10 and 5.7.11) in the network, in addition to the UPnP Discovery and Control requirements as specified in 7.2.5.5.27.6.

~~14.3.2.1.1~~7.8.2.1.1

[GUIDELINE] DLNA device classes and capabilities that implement the following device functions shall fully support the applicable mandatory portions of CEA-2014-A [101]: RUIS, RUIC, RUIS-CP, RUIC-CP, UITS, UITC, UI Content and RUI Presenter.

[ATTRIBUTES]

M	A	+RUISRC+ +RUISINK+ +RUIPL+ +RUICTRL+	n/a	n/a	n/a	AJVSR

NOTE: The device functions and applicable device capabilities are described in more detail in clause 5, and in the remainder of this subclause.

~~14.3.2.1.2~~7.8.2.1.2

[GUIDELINE] A RUI Source capability shall implement a level 1 or level 2 remote UI server as defined in [101], 5.1.1.2 and 5.1.1.3.

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	7HERE

NOTE: This includes support for all requirements as defined in [101] for a 2-Box and 3-Box remote UI server.

~~14.3.2.1.3~~7.8.2.1.3

[GUIDELINE] A RUI Pull Controller capability shall implement a UI control point as defined in [101], 5.1.1.6, and a level 0 remote UI client as defined in [101], 5.1.2.1.

[ATTRIBUTES]

M	A	+RUIPL+	n/a	n/a	n/a	R8S6C	

NOTE: This includes support for all requirements as defined in [101] for a 2-Box remote UI client.

~~14.3.2.1.4~~7.8.2.1.4

[GUIDELINE] A RUI Sink capability shall implement a level 1 or level 2 remote UI client as defined in [101], 5.1.2.

[ATTRIBUTES]

M	A	+RUISINK+	n/a	n/a	n/a	GV8WJ	

NOTE: This includes support for all requirements as defined in [101] for a 3-Box remote UI client.

~~14.3.2.1.5~~7.8.2.1.5

[GUIDELINE] A RUI Controller capability shall implement a UI control point as defined in subclauses [101], 5.1.1.6 and 5.1.2.4.

[ATTRIBUTES]

M	A	+RUICTRL+	n/a	n/a	n/a	JVSRF	

NOTE: A RUI Controller capability supports both RUI client control point requirements and RUI server control point requirements.

~~14.3.2.2~~7.8.2.2 UI Transport and connection management

This subclause defines the transport mechanism for remote UI and how a remote UI connection is initiated and terminated.

~~14.3.2.2.1~~7.8.2.2.1

[GUIDELINE] A RUI Pull controller shall implement the "CE-HTML-1.0" protocol as defined by [101]. Specifically, the RUI Pull Controller shall implement requirement [Req. 5.1.2.c] and all requirements for a 2-Box remote UI client in [101], 5.2 through 5.10.

[ATTRIBUTES]

M	A	+RUIPL+	n/a	n/a	n/a	HERET	

~~14.3.2.2.2~~7.8.2.2.2

[GUIDELINE] A RUI Pull Controller should implement 'CE-HTML-1.0_SAVED' protocol. Specifically, the RUI Pull Controller Capability should implement requirements in [101], 5.8.

[ATTRIBUTES]

S	A	+RUIPL+	n/a	n/a	n/a	8S6C7	

~~14.3.2.2.3~~7.8.2.2.3

[GUIDELINE] A RUI Source Capability shall implement the "CE-HTML-1.0" protocol. Specifically, RUI Source Capability shall implement requirements [Req 5.1.1.5.a], [Req 5.1.1.5b], [Req 5.1.1.5.c] and [Req 5.1.1.5.d] and all requirements for a 2-Box and 3-Box remote UI server in subclauses [101], 5.2 through 5.10.

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	V8WJB	

~~14.3.2.2.4~~7.8.2.2.4

[GUIDELINE] A RUI Source Capability should implement 'CE-HTML-1.0_SAVED' protocol. Specifically, the RUI Source Capability should implement requirements in [101], 5.8.

[ATTRIBUTES]

S	A	+RUISRC+	n/a	n/a	n/a	ERETD	

~~14.3.2.2.5~~7.8.2.2.5

[GUIDELINE] A RUI Sink Capability shall implement the "CE-HTML-1.0" protocol. Specifically, RUI Sink Capability shall implement requirements [Req. 5.1.2.c] and all requirements for a 3-Box remote UI client in [101], 5.2 through 5.10.

[ATTRIBUTES]

M	A	+RUISINK+	n/a	n/a	n/a	8WJBQ	

~~14.3.2.2.6~~7.8.2.2.6

[GUIDELINE] A RUI Sink Capability should implement 'CE-HTML-1.0_SAVED' protocol. Specifically, the RUI Sink Capability should implement requirements in [101], 5.8.

[ATTRIBUTES]

S	A	+RUISINK+	n/a	n/a	n/a	S6C7L	

~~14.3.2.2.7~~7.8.2.2.7

[GUIDELINE] If a RUI Pull controller sets up a connection to a remote UI for which the shortName-attribute of the <protocol>-element in the XML UI Listing has value "CE-HTML-1.0", then the RUI Pull controller shall follow [101], Req. 5.2.2.a and Req. 5.2.2.b, regarding RUI profile matching and shall follow the HTTP header requirements as defined by [101], 5.3, whilst issuing an HTTP-GET request to one of the URLs listed by a <uri>-child-element of the <protocol>-element.

[ATTRIBUTES]

M	A	+RUIPL+	n/a	n/a	n/a	VSRFA	

NOTE: The XML UI Listing which this requirement refers to is defined in [101], 5.1.1.5. See also 7.8.2.3 about profile matching.

~~14.3.2.2.8~~7.8.2.2.8

[GUIDELINE] A RUI Sink capability shall support the set up, control and termination of remote UI connections as defined in [101], 5.1.2.

[ATTRIBUTES]

M	A	+RUISINK+	n/a	n/a	n/a	SRFA9	

~~14.3.2.2.9~~7.8.2.2.9

[GUIDELINE] A RUI controller capability shall support the set up, control and termination of remote UI connections as defined in [101], 5.1.2.4.

[ATTRIBUTES]

M	A	+RUICTRL+	n/a	n/a	n/a	RETD2	

~~14.3.2.2.10~~7.8.2.2.10

[GUIDELINE] A RUI Source capability shall be able to tolerate a termination of a remote UI connection by a RUI Pull controller or RUI Sink capability closing any open TCP/IP connection at any point in time.

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	6C7LY	

NOTE: There is no explicit protocol message for terminating the remote UI connection. A RUI Pull controller can terminate the remote UI connection at any point in time by closing any open TCP/IP connection related to the remote UI. A RUI source can clean up the resources after a RUI source's defined timeout period in which no subsequent HTTP or other requests have been made related to a particular remote UI instance. A RUI Pull controller still needs to close TCP/IP connections related to the Notifsocket mechanism as defined in 5.5.1 of [101] (see also 7.8.2.5 below).

The RUI source can offer some means by which a RUI Pull controller and RUI Sink capability can continue their remote UI session after the RUI source's timeout period. This can be done for example by cookies, supporting the "CE-HTML-1.0_SAVED" protocol as defined by [101], 5.8, or persistent storage of the session information.

~~14.3.2.3~~7.8.2.3 RUI Profile Matching

This subclause defines the mechanisms by which:

- a control point (i.e. RUI Pull controller or 3-box RUI controller) can find remote UIs offered by server (i.e. RUI source capability) that match the capability profile of the remote UI client
- a RUI source capability can filter out remote UIs based on the capabilities of the remote UI client
- a 3-box RUI controller can find remote UI clients that match the capabilities needed to render a remote UI offered by a remote UI server.

It also defines the mandatory UI profiles to be supported.

~~14.3.2.3.1~~7.8.2.3.1

[GUIDELINE] A RUI Source capability shall support [Req. 5.1.1.5.d] through [Req. 5.1.1.5.f], [Req. 5.2.1.e] through [Req. 5.2.1.g], and [Req. 5.2.2.c] through [Req. 5.2.2.f] and [Req. 5.2.2.i] of [101] regarding profile matching.

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	WJBQZ	

NOTE: If a RUI Source capability serves A/V content as part of the UI, it must adhere to the A/V capability matching requirements defined in 7.8.2.6 about combining RUI with A/V.

~~14.3.2.3.2~~7.8.2.3.2

[GUIDELINE] A RUI Sink capability shall support [Req. 5.1.2.2.a], [Req. 5.1.2.2.c], [Req. 5.1.2.3.c], [Req. 5.2.1.a] through [Req. 5.2.1.d], [Req. 5.2.2.a], [Req. 5.2.2.b], [Req. 5.2.2.d] and [Req. 5.2.2.j] of [101] regarding profile matching.

[ATTRIBUTES]

M	A	+RUISINK+	n/a	n/a	n/a	RFA96	

NOTE: If a RUI Sink capability supports the rendering of A/V content as part of the UI, it must adhere to the A/V capability matching requirements defined in 7.8.2.6 about combining RUI with A/V.

~~14.3.2.3.3~~7.8.2.3.3

[GUIDELINE] A RUI Pull Controller shall support [Req. 5.2.1.a] through [Req. 5.2.1.d], [Req. 5.2.2.a], [Req. 5.2.2.b], [Req. 5.2.2.d] and [Req. 5.2.2.j] of [101] regarding profile matching.

[ATTRIBUTES]

M	A	+RUIPL+	n/a	n/a	n/a	ETD2Z	

NOTE: If a RUI Pull Controller supports the rendering of A/V content as part of the UI, it must adhere to the A/V capability matching requirements defined in 7.8.2.6 about combining RUI with A/V.

~~14.3.2.3.4~~7.8.2.3.4

[GUIDELINE] If a RUI Pull Controller or RUI Sink capability discovers through capability matching that there is no match for a particular UI, then an indication shall be given to the user of the following condition at the time remote UIs are exposed for user selection:

- The selected rendering endpoint is not capable of rendering the remote UI.

The form of this user indication is implementation dependent and can be user selectable.

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	C7LY6	

NOTE: This guideline is the Remote UI analogous requirement to guideline 6.1.4.1 of [56]. Note that a RUI Controller capability (+RUICTRL+) does not need to follow this requirement, since it can select the remote UI before selecting a target RUI Sink capability.

~~14.3.2.3.5~~7.8.2.3.5

[GUIDELINE] A RUI Source capability should tailor its specific UI based on the A/V capabilities reported by a connecting RUI Pull Controller or RUI Sink capability, so as to inform the user if a given content item cannot be rendered by the A/V rendering device.

[ATTRIBUTES]

S	A	+RUISRC+	n/a	n/a	n/a	JBQZU	

~~14.3.2.3.6~~7.8.2.3.6

[GUIDELINE] Every RUI Source capability shall support [Req. 5.2.1.f] of [101], which states that a RUI Source shall at a minimum support the UI profile "SD_UIPROF".

Support here means that the RUI Source capability shall list at least one <ui> in its XML UI Listing (as defined in subclause 5.1.1.5 of [101]) that has at least one <uri> that points to CE-HTML content adhering to the SD_UIPROF profile, without requiring extensions, and for which the <profilelist> element in the XML UI Listing that corresponds to that <uri> contains a <ui_profile>-element with SD_UIPROF as value for the "name" attribute.

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	FA96V	

~~14.3.2.3.7~~7.8.2.3.7

[GUIDELINE] If a RUI Source capability exposes a remote UI for a UI profile that is unequal to "SD_UIPROF", then the RUI Source capability should also expose a remote UI with similar functionality for the "SD_UIPROF" profile.

[ATTRIBUTES]

S	A	+RUISRC+	n/a	n/a	n/a	TD2ZH	

~~14.3.2.3.8~~7.8.2.3.8

[GUIDELINE] If a RUI Pull Controller or a RUI Sink capability is co-located with a Device Class of the HND Device Category, then it shall at a minimum support the UI profile "SD_UIPROF".

Support here means that the RUI Pull Controller and RUI Sink capability shall at least be able to render the remote UI and have the ability to interact with it according to the minimum device characteristics as defined for the SD_UIPROF in Table 4 in subclause 5.2.1 of [101].

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	7LY67	

~~14.3.2.3.9~~7.8.2.3.9

[GUIDELINE] If a RUI Pull Controller or a RUI Sink capability is co-located with a Device Class of the MHD Device Category, then it shall at a minimum support the UI profile "MD_UIPROF".

Support here means that the RUI Pull Controller and RUI Sink capability shall at least be able to render the remote UI and have the ability to interact with it according to the minimum device characteristics as defined for the MD_UIPROF in Table 4 in subclause 5.2.1 of [101].

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	BQZU3	

~~14.3.2.3.10~~7.8.2.3.10

[GUIDELINE] A RUI Pull controller and RUI Sink capability should also support other UI profiles than the mandatory UI profile as specified for its Device Category in 7.8.2.3.8 and 7.8.2.3.9.

[ATTRIBUTES]

S	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	85BQD	

~~14.3.2.4~~7.8.2.4 UI Updates

This subclause defines the mechanisms for dynamically updating the UI with new information coming from the RUI Source capability.

~~14.3.2.4.1~~7.8.2.4.1

[GUIDELINE] A RUI Pull Controller and a RUI Sink capability shall support the NotifSocket mechanism as defined in subclause 5.5.1 of [101].

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	R8UDU	

~~14.3.2.4.2~~7.8.2.4.2

[GUIDELINE] A RUI Pull Controller and a RUI Sink capability should support the XMLHttpRequest mechanism as defined in subclause 5.5.2 of [101], but may refuse to process notifications that are being polled for too regularly, i.e. if the RUI Pull Controller or RUI Sink capability is being requested to process asynchronous XML updates within less than 1 second intervals.

[ATTRIBUTES]

S	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	QSVVY	

NOTE: +RUISRC+ capability implementations need only consider the use of XMLHttpRequest for meta-data driven applications (such as fetching EPG XML data that is displayed inside the UI) and/or Internet based applications.

~~14.3.2.4.3~~7.8.2.4.3

[GUIDELINE] A RUI Source capability should not use the XMLHttpRequest mechanism with a polling interval of less than 1 second.

[ATTRIBUTES]

S	A	+RUISRC+	n/a	n/a	n/a	SZ7SW	

NOTE: +RUISRC+ capability implementations need only consider the use of XMLHttpRequest for meta-data driven applications (such as fetching EPG XML data that is displayed inside the UI) and/or Internet based applications.

Note: this guideline can be tested by measuring the polling interval between two or more consecutive XMLHttpRequests. If this is less than 1 second, then it doesn't support this guideline.

~~14.3.2.4.4~~7.8.2.4.4

[GUIDELINE] A RUI Source Capability should not use those TCP/UDP port numbers for UI updates which are reserved by other DLNA guidelines.

[ATTRIBUTES]

S	A	+RUISRC+	n/a	n/a	n/a	5BQDR	

NOTE: Some port numbers greater than 1024 have been reserved in DLNA guidelines for specific use. For example, port 1900 is reserved for multicast discovery messages.

~~14.3.2.5~~7.8.2.5 Third Party Notifications

~~14.3.2.5.1~~7.8.2.5.1

[GUIDELINE] A RUI Sink capability and a RUI Pull Controller capability shall be capable of receiving multicast notifications defined in [Req. 5.6.1a] of [101].

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	SVVYG	

~~14.3.2.5.2~~7.8.2.5.2

[GUIDELINE] A RUI Source capability with multicast notification enabled shall implement the mechanism defined in subclause 5.6.1 of [101].

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	Z7SWJ	

~~14.3.2.5.3~~7.8.2.5.3

[GUIDELINE] A RUI Sink capability and a RUI Pull Controller capability shall be capable of handling the multicast notification related protocol as defined in [Req. 5.6.3a] and [Req. 5.6.3c] of [101].

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	BQDRF	

~~14.3.2.5.4~~7.8.2.5.4

[GUIDELINE] A RUI Sink capability and a RUI Pull Controller capability shall be capable of displaying the multicast notification content as defined in [Req. 5.6.3d] through [Req. 5.6.3i] of [101].

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	VVYGX	

~~14.3.2.5.5~~7.8.2.5.5

[GUIDELINE] A RUI Source capability with multicast notification enabled shall be capable of handling the multicast notification related protocol as defined in [Req. 5.6.3b] and [Req. 5.6.3f] of [101].

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	7SWJW	

~~14.3.2.6~~7.8.2.6 Combining RUI with A/V

~~14.3.2.6.1~~7.8.2.6.1 General

The requirements in this subclause are based on the assumption that the remote UI rendering endpoint is the same as the A/V rendering endpoint.

~~14.3.2.6.2~~7.8.2.6.2 A/V Object Support

~~14.3.2.6.2.1~~7.8.2.6.2.1

[GUIDELINE] If a Rendering Endpoint that supports the Audio Media Class implements a RUI Pull Controller or RUI Sink capability, then it shall support requirements [Req. 5.7.1.a], [Req. 5.7.1.b], and [Req. 5.7.1.f] related to the inclusion, control and rendering of "audio" content/objects in subclause 5.7.1 of [101].

[ATTRIBUTES]

M	A	DMP DMR	M-DMP	n/a	n/a	QDRFS	

~~14.3.2.6.2.2~~7.8.2.6.2.2

[GUIDELINE] If a RUI Pull Controller or RUI Sink capability indicates support for a DLNA Media Profile of the Audio Media Class, by including an <audio_profile>-element in its capability description (as defined by [Req. 5.2.1.a] and [Req. 5.2.1.c] of [101]), then it shall support requirements [Req. 5.7.1.a], [Req. 5.7.1.b], and [Req. 5.7.1.f] related to the inclusion, control and rendering of "audio" content/objects in subclause 5.7.1 of [101].

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	DULWG	

~~14.3.2.6.2.3~~7.8.2.6.2.3

[GUIDELINE] If a Rendering Endpoint that supports the AV Media Class implements a RUI Pull Controller or RUI Sink capability, then it shall support all requirements in subclauses 5.7.1 and 5.7.3 of [101].

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	VYGXQ	

~~14.3.2.6.2.4~~7.8.2.6.2.4

[GUIDELINE] If a RUI Pull Controller or RUI Sink capability indicates support for a DLNA Media Profile of the AV Media Class, by including an <video_profile>-element in its capability description (as defined by [Req. 5.2.1.a] and [Req. 5.2.1.c] of [101]), then it shall support all requirements in subclause 5.7.1 and 5.7.3 of [101].

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	SWJWS	

~~14.3.2.6.3~~7.8.2.6.3 A/V Capability Matching

~~14.3.2.6.3.1~~7.8.2.6.3.1

[GUIDELINE] If a RUI Source capability uses the A/V plugin-object (as defined in subclause 5.7.1 of [101]) in its CE-HTML content to offer content of the Audio and/or A/V Media Class, then the RUI Source shall support the DLNA Media Format profiles that are defined as mandatory for that Media Class, as defined in "DLNA Guidelines Part 2: Media Format Profiles", whereby the Device Category shall be the same as the Device Category of the Device Class to which the RUI Source capability is added.

Support here means that the RUI Source shall have at least one <ui> in its XML UI Listing (as defined in subclause 5.1.1.5 of [101]) that has at least one <uri> that points, either directly or through linking, to CE-HTML content that includes an A/V plugin object that links to content of such mandatory Media Format.

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	DRFSN	

NOTE: This guideline is the Remote UI analogous requirement to Requirements 6.1.1.2 of "DLNA Guidelines Part 2: Media Format Profiles". Here's an example to illustrate this requirement: if a RUI Source capability is added to an HND device (such as a DMS), then if any audio is offered through the RUI Source capability, it must at least offer one remote UI that supports LPCM audio content.

~~14.3.2.6.3.2~~7.8.2.6.3.2

[GUIDELINE] If a RUI Source exposes an A/V content item encoded to an optional DLNA media format profile through its remote UI, then the RUI Source should also advertise the content item encoded to a mandatory DLNA media format profile.

[ATTRIBUTES]

S	A	+RUISRC+	n/a	n/a	n/a	ULWGU	

NOTE: This guideline is the Remote UI analogous requirement to Requirement 6.1.3.1 of "DLNA Guidelines Part 2: Media Format Profiles". There are various number of ways to meet this recommendation, for example by nesting <object>-elements as defined by bullet 4 of [Req. 5.7.1.b] of [101].

~~14.3.2.6.3.3~~7.8.2.6.3.3

[GUIDELINE] If a RUI Pull Controller or a RUI Sink capability implements the A/V plugin-object as defined in subclause 5.7.1 of [101], then the RUI Pull Controller and RUI Sink capability shall support the same mandatory DLNA Media Format profiles for an A/V plugin object as are supported for the Device Category of the Device Class to which the RUI Pull Controller or RUI Sink capability is added.

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	YGXQ5	

NOTE: Here's an example to illustrate this requirement: if a RUI Pull Controller is added to an HND device (such as a DMP), then if in Europe video is supported for the A/V plugin object, it must at least support MPEG2_PS_PAL. The mandatory formats are defined in subclause 6.2 of the "DLNA Guidelines Part 2: Media Format Profiles".

~~14.3.2.6.3.4~~7.8.2.6.3.4

[GUIDELINE] If a RUI Pull Controller or a RUI Sink capability implements the A/V plugin-object as defined in subclause 5.7.1 of [101], then it should support the same DLNA Media Format profiles for playback inside a CE-HTML page as the Device Class, to which the RUI Pull Controller or RUI Sink capability is added, supports for playback using DLNA Media Transport.

[ATTRIBUTES]

S	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	WJWSZ

NOTE: So if the RUI Pull controller or RUI Sink capability is added to a DMP or DMR, it is encouraged to support at least the same media profiles for playback inside a CE-HTML page as supported by the DMP or DMR for non-RUI purposes.

~~14.3.2.6.3.5~~7.8.2.6.3.5

[GUIDELINE] If a RUI Source capability uses the A/V plugin-object (as defined in subclause 5.7.1 of [101] in its CE-HTML content) to offer content of the Audio and/or A/V Media Class, then it shall list the DLNA media profiles that are required to be supported by a RUI Pull Controller or RUI Sink capability to play the A/V content, by adding one or more corresponding <audio_profile> or <video_profile> elements inside the <profilelist>-element of the Remote UI Server Capability Description of the corresponding UI in the XML UI Listing as defined by [Req. 5.1.1.5.d] through [Req. 5.1.1.5.f] of [101].

"Corresponding <audio_profile>or <video_profile>" here means that:

a) If the content-type as denoted through the value for attribute "type" of the <object>-element correspond to the MIME-type of one of the DLNA media formats (as defined in "DLNA Guidelines Part 2: Media Format Profiles" [56]), then that content-type shall be listed through the "type"-attribute of an <audio_profile> or <video_profile>-element in the Remote UI Server Capability Description. The corresponding DLNA Media profile name shall be provided either through a value of the "name"-attribute, or through one of the profiles listed by the "alt"-attribute of the <audio_profile> or <video_profile>-element

b) If the transport protocol (as denoted through the URI-scheme used for the URI value of the "data" attribute of the <object>-element) is one of the DLNA supported URI schemes: "http://", "rtsp://" or "rtspu://", then a corresponding transport type shall be specified through the "transport"-attribute of an <audio_profile> or <video_profile>-element in the remote UI server capability description, in the following manner:

- a case-insensitive value of "DLNA_HTTP" corresponds to the "http://" URI scheme
- a case-insensitive value of "DLNA_RTP" corresponds to any of the following URI schemes: "rtsp://" or "rtspu://".

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	RFSNW

NOTE: An example: if a PAL MPEG-2 Program Stream is offered in a DLNA compliant way through the A/V plugin object using

```
<object data="http://10.0.0.1/video.mpg" type="video/mpeg"/>
```

then the following must be added to the Remote UI Server Capability Description:

```
<video_profile type="video/mpeg" name="MPEG_PS_PAL" transport="dlna_http"/>
```

~~14.3.2.6.3.6~~7.8.2.6.3.6

[GUIDELINE] If a RUI Pull Controller or a RUI Sink capability implements the A/V plugin-object as defined in subclause 5.7.1 of [101], then it shall list the supported DLNA Media Format profiles for playback and control inside a CE-HTML page, by adding the corresponding <audio_profile> or <video_profile> elements inside the <profilelist>-element of the Remote UI Client Capability description (as defined in subclause 5.2 of [101]).

"Corresponding <audio_profile>or <video_profile>" here means that:

a) The MIME-type of the supported DLNA Media Format profile (as defined in "DLNA Guidelines Part 2: Media Format Profiles" [56]) shall be listed through the "type" attribute, and the DLNA Media Format profile name shall be provided through the "name"-attribute of an <audio_profile> or <video_profile>-element.

b) The "transport" attribute of the <audio_profile> or <video_profile>-element shall be given:

- a case-insensitive value of "DLNA_HTTP" in case of HTTP streaming
- a case-insensitive value of "DLNA_RTP" in case of RTP streaming

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	LWGU6	

~~14.3.2.6.4~~7.8.2.6.4 **A/V Metadata and getSinkProtocolInfo**

~~14.3.2.6.4.1~~7.8.2.6.4.1

[GENERAL] For the playback of A/V inside the remote UI we cannot rely on the CDS <res>-element to provide the A/V metadata, in particular the 4th field protocolInfo-values that are needed to properly set up and use a DLNA compatible A/V stream. CEA-2014-A ([101])defines a parameter called "dlna_res_attr" that can be set for the A/V plugin object through a <param>-element. This parameter is the remote UI equivalent to the <res>-element, and can be used as a mechanism to provide the necessary A/V metadata to set up and use a DLNA compatible A/V stream, in addition to the HTTP header "contentFeatures.dlna.org" (as defined in; see also 7.8.2.6.5.2 and 7.8.2.6.5.3). This subclause defines the requirements related to the "dlna_res_attr", and in particular the requirements related to res@protocolInfo, including the requirements related to the getSinkProtocolInfo Javascript method of the A/V plugin object.

~~14.3.2.6.4.2~~7.8.2.6.4.2

[GUIDELINE] If a RUI Source capability includes a <param>-element "dlna_res_attr" for the A/V plugin-object (as defined in [101], 5.7.1) in its CE-HTML content, it shall support all mandatory guidelines for the attributes of the <res>-element as defined in 7.4

Specifically, support for these requirements means that:

a) The following requirements shall be supported without change::

7.4.1.3.4	7.4.1.3.3.2	7.4.1.3.3.4	n/a	7.4.1.3.5.1	7.4.1.3.8
7.4.1.3.12.1	7.4.1.3.12.2	7.4.1.3.15.3	7.4.1.3.15.5	7.4.1.3.16.1	7.4.1.3.16.2
7.4.1.3.16.3	7.4.1.3.17	7.4.1.3.18.1	7.4.1.3.18.2	7.4.1.3.18.3	7.4.1.3.19.1
7.4.1.3.19.2	7.4.1.3.19.3	7.4.1.3.19.4	7.4.1.3.19.5	7.4.1.3.19.6	7.4.1.3.20.1
7.4.1.3.20.2	7.4.1.3.21.1	7.4.1.3.21.2	7.4.1.3.21.3	7.4.1.3.22.1	7.4.1.3.22.2
7.4.1.3.22.4	7.4.1.3.23.1	7.4.1.3.23.2	7.4.1.3.23.3	7.4.1.3.24	7.4.1.3.25
7.4.1.3.26.1	7.4.1.3.26.2	7.4.1.3.27.1	7.4.1.3.28.1	7.4.1.3.29	7.4.1.3.30
7.4.1.3.32.1	7.4.1.3.32.2	7.4.1.3.32.3	7.4.1.3.33.1	7.4.1.3.34.1	7.4.1.3.34.2
7.4.1.3.35.2	7.4.1.3.37	7.4.1.4.8.1	7.4.1.4.8.2		

b) The following requirements need to be re-interpreted in the following manner:

- 7.4.1.3.15.1 and 7.4.1.3.15.2: the URL value for the <res>-element in this case refers the "data"-attribute of the <object>-element for which the "dlna_res_attr" <param>-element has been defined
- 7.4.1.3.15.4: "UPnP AV Endpoints (devices and control points)" needs to be replaced with "dlna_res_attr"
- 7.4.1.3.18.4 and 7.4.1.3.18.5: "CDS object" needs to be replaced with "A/V object inside a CE-HTML page"
- 7.4.1.3.31.2: "Source arguments … UPnP AV MediaServer" needs to be replaced with "dlna_res_attr"
- 7.4.1.4.3 and 7.4.1.4.4: "CDS object (identified through an <item> or <container> element)" needs to be replaced with "A RUI Source capability", and "<res>-elements" needs to be replaced with "nested A/V objects"

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	GXQ5Y

NOTE: Basically, this requirement defines that meta-information, such as the protocolInfo can be provided to an A/V object inside the remote UI in a manner that is similar to a CDS <res>-element, i.e.:

```
<res duration="x:y:z" protocolInfo="http-get:*:video/mpeg: DLNA_ORG_PN=MPEG_PS_NTSC">URI</res>
```

can be mapped in an equivalent way to:

```
<object data="URI" type="video/mpeg">
    <param name="dlna_res_attr"
        value="duration=&quot;x:y:z&quot;
        protocolInfo=&quot;http-get:*:video/mpeg:
        DLNA.ORG_PN=MPEG_PS_NTSC&quot;"/>
</object>
```

Note that the requirements for the attributes of the <res>-element related to content upload to a DMS (such as 7.4.1.3.26.3) are not included, since these do not apply in case of remote UI. It also does not include requirements that are not related to Streaming transfers, since these are not relevant in case of the A/V plugin object.

~~14.3.2.6.4.3~~7.8.2.6.4.3

[GUIDELINE] If a RUI Source capability exposes a res@protocolInfo through the <param>-element "dlna_res_attr" for the A/V plugin-object as defined in [101], 5.7.1, in its CE-HTML content, then the context of the protocolInfo shall be to describe the content and/or transport layer features for the "data" URI value for the same instance of the <object>-element for which the <param>-element has been defined.

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	JWSZJ

NOTE: This requirement is related to the requirements of 7.4.1.3.12. Whereas that subclause covers the requirements related to UPnP A/V, this additional requirement needs to be added to apply for the remote UI case. However, instead of doing it in 7.4.1.3.12, we include the requirement in this subclause related to RUI and A/V integration.

~~14.3.2.6.4.4~~7.8.2.6.4.4

[GUIDELINE] If a RUI Source capability exposes a res@protocolInfo through the <param>-element "dlna_res_attr" for the A/V plugin-object as defined in [101], 5.7.1, in its CE-HTML content, and the value for the "data" attribute of the A/V plugin-object starts with "http://", then the 4th field parameter values shall be identical to the 4th field parameter values that can be retrieved through the getContentFeatures.dlna.org HTTP header (as defined in7.5.4.3.2.10) for the same URI as specified by the "data"-attribute of the <object>-element for which the <param>-element has been defined, at least at the time of exposure.

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	FSNWV	

NOTE: The HTTP response to a getContentFeatures.dlna.org HTTP request provides (through HTTP header "contentFeatures.dlna.org") the latest information regarding the 4th field parameter values. It is the responsibility of the author of the CE-HTML page to make sure the information provided through the "dlna_res_attr" <param>-element is consistent with the information that can be retrieved through getContentFeatures.dlna.org. Authors can use one of the RUI page update mechanisms as defined in 7.8.2.4 to update the information for the "dlna_res_attr" <param>-element.

~~14.3.2.6.4.5~~7.8.2.6.4.5

[GUIDELINE] If a RUI Source capability exposes a res@protocolInfo through the <param>-element "dlna_res_attr" for the A/V plugin-object as defined in subclause 5.7.1 of [101] in its CE-HTML content, and the value for the "data" attribute of the A/V plugin-object starts with "rtsp://" or "rtspu://", ", then the 4th field parameter values shall be identical to the DLNA contentFeatures field (a=contentFeatures.dlna.org) included in the session description section of SDP description (as defined in 7.5.4.4.6.2.66), for the same URI as specified by the "data"-attribute of the <object>-element for which the <param>-element has been defined, at least at the time of exposure.

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	WGU6Y	

NOTE: The RTSP response to a RTSP DESCRIBE request from the client to the server provides (through the "contentFeatures.dlna.org" field of the RTSP DESCRIBE response) the latest information regarding the 4th field parameter values. It is the responsibility of the author of the CE-HTML page to make sure the information provided through the "dlna_res_attr" <param>-element is consistent with the information that can be retrieved through the contentFeatures.dlna.org field in the SDP description. Authors can use one of the RUI page update mechanisms as defined in7.8.2.4 to update the information for the "dlna_res_attr" <param>-element.

~~14.3.2.6.4.6~~7.8.2.6.4.6

[GUIDELINE] If a RUI Source capability includes a <param>-element "dlna_res_attr" for the A/V plugin-object as defined in subclause 5.7.1 of [101] in its CE-HTML content, then it shall list the <audio_profile> and <video_profile> elements that correspond to what is stated in the "dlna_res_attr" inside the <profilelist>-element of the Remote UI Server Capability Description of the corresponding UI in the XML UI Listing as defined by [Req. 5.1.1.5.d] through [Req. 5.1.1.5.f] of [101].

"Corresponding <audio_profile>or <video_profile>" here means that:

a) The 'transport" attribute of the <audio_profile> or <video_profile>-element shall be given:

- the case-insensitive value "dlna_http", if the first field of the protocolInfo parameter inside the value for the "dlna_res_attr" is defined as "http-get"
- the case-insensitive value "dlna_rtp", if the first field of the protocolInfo parameter inside the value for the "dlna_res_attr" is defined as "rtsp-rtp-udp"

b) The "type" attribute of the <audio_profile> or <video_profile>-element shall be given the same MIME-type (corresponding to one of the supported DLNA Media Format profiles) as defined by the third field of the protocolInfo parameter inside the value for "dlna_res_attr".

c) Either the "name" attribute or the "alt" attribute of the <audio_profile> or <video_profile>-element shall contain the DLNA Media Format profile name as defined by the DLNA.ORG_PN parameter as defined by the fourth field of the protocolInfo parameter inside the value for "dlna_res_attr".

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	XQ5YY	

NOTE: Here are some examples of <audio_profile> and <video_profile> elements corresponding to a protocolInfo attribute defined inside the value of the "dlna_res_attr" <param>-element:

- <audio_profile transport="dlna_http" name="DLNA_PROFILE"> corresponds to protocolInfo value "http-get:*:${mime-type for DLNA_PROFILE}:DLNA.ORG_PN=${DLNA_PROFILE}"
- <video_profile transport="dlna_rtp" name="DLNA_PROFILE"> corresponds to protocolInfo value "rtsp-rtp-udp:*:${mime-type for DLNA_PROFILE}:DLNA.ORG_PN=${DLNA_PROFILE}"

Whereby ${..} needs to be replaced with actual value

~~14.3.2.6.4.7~~7.8.2.6.4.7

[GUIDELINE] The method getSinkProtocolInfo() of the A/V plugin-object as defined in [101], 5.7.1, shall list the union set of protocolInfo values that correspond to the DLNA Media Format profiles that are supported for playback through the A/V plugin object inside a CE-HTML page.

This shall be done in accordance with the following requirements:

a) The following requirements shall be supported without change:

7.4.1.3.15.3	7.4.1.3.16.1	7.4.1.3.16.2	7.4.1.3.16.3	7.4.1.3.17	7.4.1.3.18.1
7.4.1.3.21.1	7.4.1.3.21.3	7.4.1.3.22.1	7.4.1.3.22.2	7.4.1.3.22.4	7.4.1.3.23.1
7.4.1.3.23.2	7.4.1.3.23.3	7.4.1.3.25	7.4.1.3.26.1	7.4.1.3.26.2	7.4.1.3.31.1

b) The following requirements need to be re-interpreted in the following manner:

- 7.4.1.3.14.2: "CMS:GetProtocolInfo" needs to be replaced with "method getSinkProtocolInfo() of the A/V plugin object"
- 7.4.1.3.14.3: "ConnectionManager service" needs to be replaced with "method getSinkProtocolInfo() of the A/V plugin object"
- 7.4.1.3.15.4: "UPnP AV Endpoints (devices and control points)" needs to be replaced with "method getSinkProtocolInfo() of the A/V plugin object"
- 7.4.1.3.27.3 and 7.4.1.3.33.2: "SinkProtocolInfo values … UPnP AV MediaRenderer" needs to be replaced with "protocolInfo values returned by method getSinkProtocolInfo() of the A/V plugin object"
- 7.4.1.3.32.4: "Sink argument's …UPnP AV MediaRenderer" needs to be replaced with "protocolInfo values returned by method getSinkProtocolInfo() of the A/V plugin object"
- 7.4.1.6.1.1: "UPnP AV MediaRenderer … its SinkProtocolInfo" needs to be replaced with "RUI Pull Controller or RUI Sink capability that returns a protocolInfo for method getSinkProtocolInfo() of the A/V plugin object", and "SinkProtocolInfo refers …", and the comment below the requirement needs to be ignored.
- 7.4.1.6.1.2: "UPnP AV MediaRenderer" needs to be replaced with "RUI Pull Controller or RUI Sink capability", and "SinkProtocolInfo (as observed …" needs to be replaced with "protocolInfo values returned by method getSinkProtocolInfo() of the A/V plugin object".

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	U5BVY	

~~14.3.2.6.4.8~~7.8.2.6.4.8

[GUIDELINE] If a RUI Pull controller or RUI Sink capability lists a protocolInfo value through the getSinkProtocolInfo() for the A/V plugin-object as defined in subclause 5.7.1 of [101], then the context of the protocolInfo shall be to describe the ability of the device hosting the RUI Pull controller or RUI Sink capability to support the indicated feature or content. See 7.4.1.3.12.8 for more details about supporting a feature.

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	WSZJX	

NOTE: The requirements of 7.4.1.3.12 are only written for UPnP A/V. This additional requirement needs to be added to apply for the remote UI case. However, instead of doing it in 7.4.1.3.12, we include the requirement in this subclause related to RUI and A/V integration.

~~14.3.2.6.4.9~~7.8.2.6.4.9

[GUIDELINE] The union set of protocolInfo values returned by method getSinkProtocolInfo() as denoted by 7.8.2.6.4.7 shall include the list of DLNA media formats listed through the <audio_profile> and <video_profile> elements in the remote UI Client Capability Description as sent by a RUI Pull Controller or RUI Sink capability in the "User-Agent" header while connecting to the RUI Source for the CE-HTML page that calls the getSinkProtocolInfo() method

If it reports support for one or more DLNA media format profiles, it shall explicitly list each media profile separately using protocolInfo values, whereby:

a) A case-insensitive value "dlna_http" defined as value for the 'transport' attribute of the <audio_profile> or <video_profile>-element corresponds to a first field protocolInfo value "http-get".
b) Acase-insensitive value "dlna_rtp" defined as value for the 'transport' attribute of the <audio_profile> or <video_profile>-element corresponds to a first field protocolInfo value "rtsp-rtp-udp".
c) The value for the "type" attribute of the <audio_profile> or <video_profile>-element corresponds to the third field protocolInfo value.
d) The value for the "name" attribute of the <audio_profile> or <video_profile>-element corresponds to the value of the DLNA.ORG_PN parameter inside the fourth field protocolInfo value.

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	SNWVP	N

NOTE: For some examples of correspondence between <audio_profile> and <video_profile> elements, and protocolInfo values, we refer to the comments of 7.8.2.6.4.6.

~~14.3.2.6.4.10~~7.8.2.6.4.10

[GUIDELINE] On a RUI Sink capability, the method getSinkProtocolInfo() of the A/V plugin-object as defined in subclause 5.7.1 of [101], shall list the same DLNA media formats as listed through the <audio_profile> and <video_profile> elements in the remote UI Client capability description as can be retrieved through the <profilelist>-element inside a level 1 or level 2 remote UI client device description hosted by a RUI Sink capability.

If the RUI Sink capability is a Level 2 Remote UI Client device (as defined in subclause 5.1.2.3 of [101]), then the method getDeviceProfile() method of the Level 2 Remote UI client service, shall list the same DLNA media formats that are listed by method getSinkProtocolInfo() of the A/V plugin-object.

[ATTRIBUTES]

M	A	+RUISINK+	n/a	n/a	n/a	GU6YQ	

NOTE: The correspondence between <audio_profile> and <video_profile>-elements and values for getSinkProtocolInfo() is the same as in 7.8.2.6.4.9.

~~14.3.2.6.4.11~~7.8.2.6.4.11

[GUIDELINE] If the RUI Pull Controller or RUI Sink capability is added to a DMR, the returned list of DLNA media formats by the getSinkProtocolInfo() Javascript method for the A/V plugin object should either be the same or a subset of the DLNA media formats as listed through the return value of the GetProtocolInfo() method of a ConnectionManager service of a DMR.

[ATTRIBUTES]

S	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	Q5YYR	

NOTE: Some A/V media format profiles might not be supported for playback as part of a CE-HTML page.

~~14.3.2.6.5~~7.8.2.6.5 Media Transport and Control

~~14.3.2.6.5.1~~7.8.2.6.5.1

[GUIDELINE] If a RUI Pull Controller or RUI Sink capability indicates support for a DLNA Media Profile by including an <audio_profile>- or <video_profile>-element in its capability description (as defined by [Req. 5.2.1.a] and [Req. 5.2.1.c] of [101]), then it shall adhere to the following requirements for setting up and controlling the A/V stream through the Remote UI:

a) If the device to which the RUI Pull Controller or RUI Sink capability is added is a Rendering Endpoint with an MSCP, then it shall adhere to subclause 7.3 and 7.4 of the DLNA guidelines for setting up and controlling the A/V stream, in the same way as would be done by such corresponding device class without remote UI.

b) In all other cases, setting up and controlling the A/V stream shall be done using the DLNA Media Transport guidelines as defined by subclause 7.4 of the DLNA guidelines..

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	NWVPG	

NOTE: Note that this requirement is about set up and control, not browsing the CDS, i.e. in both cases we don't rely on the client device to use the CDS-browse method to fetch the meta-information of the A/V stream, such as the protocolInfo. Instead this will be provided through the <param>-element "dlna-res-attr" for the A/V plugin object as defined by 7.8.2.6.4.2.

Bullet 1 of the above requirement is based on using available UPnP services (such as ConnectionManager and AVTransport service) on the device that hosts the A/V stream. This requirement can be tested by re-doing all the tests for a DMP resp. a DMC with a local DMR, that logically follow after a CDS-browse action, for one or more remote UIs that offers A/V through an A/V plugin object (as defined by subclause 5.7.1 of [101]). This includes individual A/V control actions (such as play, stop and pause) as defined below. The necessary Javascript to call these individual A/V control actions can easily be generated.

Bullet 2 of the above requirement can be tested by checking compliance with overarching 7.8.2.6.5.2 and 7.8.2.6.5.5, 7.8.2.6.5.9, 7.8.2.6.5.11, and 7.8.2.6.5.13 for the individual actions, using Javascript to test these individual A/V control actions.

~~14.3.2.6.5.2~~7.8.2.6.5.2

[GUIDELINE] If a RUI Pull Controller or RUI Sink capability indicates support for a DLNA Media Profile through an <audio_profile>- or <video_profile>-element in its capability description (as defined by [Req. 5.2.1.a] and [Req. 5.2.1.c] of [101]), then it shall support all Content Receiver related requirements in 7.5 that apply to Streaming Transfers, with the following exception:

- 7.5.4.3.2.10.7, where the first paragraph of the comment has to be amended with ‘ where the protocolInfo may also be provided through the <param>-element "dlna_res_attr" for the A/V plugin object as defined by 7.8.2.6.4.2’.

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	U6YQ9	

NOTE: The following informative list shows the resulting requirements:

7.5.4.2.1	7.5.4.2.2	7.5.4.2.3.1	7.5.4.2.3.2	7.5.4.2.3.5	7.5.4.2.4.6
7.5.4.2.5	7.5.4.2.7.1	7.5.4.2.12.1	7.5.4.2.13	7.5.4.3.2.1.1	7.5.4.3.2.1.2
7.5.4.3.2.1.3	7.5.4.3.2.2	7.5.4.3.2.3	7.5.4.3.2.4.4	7.5.4.3.2.4.6	7.5.4.3.2.4.8
7.5.4.3.2.4.11	7.5.4.3.2.5.1	7.5.4.3.2.5.2	7.5.4.3.2.5.3	7.5.4.3.2.6	7.5.4.3.2.7.5

7.5.4.3.2.8.1	7.5.4.3.2.10.3	7.5.4.3.2.10.4	7.5.4.3.2.10.7 [a]	7.5.4.3.2.11.2	7.5.4.3.2.11.3
7.5.4.3.2.15.5	7.5.4.3.2.16.1	7.5.4.3.2.16.2	7.5.4.3.2.19.1	7.5.4.3.2.19.3	7.5.4.3.2.19.6
7.5.4.3.2.20.2	7.5.4.3.2.20.3	7.5.4.3.2.20.6	7.5.4.3.2.20.10	7.5.4.3.2.20.16	7.5.4.3.2.21.1
7.5.4.3.2.21.2	7.5.4.3.2.22.3	7.5.4.3.2.22.11	7.5.4.3.2.21.2	7.5.4.3.2.22.3	7.5.4.3.2.22.11
7.5.4.3.2.24.3	7.5.4.3.2.24.11	7.5.4.3.2.24.12	7.5.4.3.2.26.1	7.5.4.3.2.26.2	7.5.4.3.2.26.3
7.5.4.3.2.26.4	7.5.4.3.2.27.1	7.5.4.3.2.28.1	n/a	7.5.4.3.2.29.1	7.5.4.3.2.30
7.5.4.3.2.31	7.5.4.3.2.33.1	7.5.4.3.2.33.3	7.5.4.3.2.33.5	7.5.4.3.2.34.1	7.5.4.3.2.35.1
7.5.4.3.2.36.1	7.5.4.3.2.36.2	7.5.4.3.2.37.1	7.5.4.3.2.37.2	7.5.4.3.2.38.2	7.5.4.3.2.38.3
7.5.4.3.3.1.1	7.5.4.3.3.2	7.5.4.3.3.3.1	7.5.4.3.3.4.1	7.5.4.3.3.4.3	7.5.4.3.3.5.1
7.5.4.3.3.5.2	7.5.4.3.3.5.3	7.5.4.3.3.6.1	7.5.4.3.3.6.2	7.5.4.3.3.7.1	7.5.4.3.3.8.1
7.5.4.3.3.9.1	7.5.4.3.3.10.1	7.5.4.3.3.11.1	7.5.4.3.3.12.1	7.5.4.3.3.12.2	7.5.4.3.3.15.1
7.5.4.3.3.16.3	7.5.4.3.3.16.8	7.5.4.3.3.20.1	7.5.4.3.3.20.7	7.5.4.3.3.21.1	

[a] whereby the protocolInfo can also be provided through the <param>-element "dlna-res-attr" for the A/V plugin object as defined by 7.8.2.6.4.2

If the RUI Pull Controller or RUI Sink capability support RTP streaming (i.e. if one or more <audio_profile> or <video_profile> elements in the capability profile has an attribute "transport" with case-insensitive value "dlna_rtp"), then this also includes all requirements related to an RTP Receiving Endpoint as indicated by 7.5.4.4.2.1 and onwards.

Note that all mandatory 7.4's requirements remain mandatory, all recommendations remain recommendations and all optional requirements remain optional unless specified otherwise in this subclause (i.e. 7.8.2.6 "Combining RUI with A/V").

~~14.3.2.6.5.3~~7.8.2.6.5.3

[Guideline] If a RUI Source capability uses the A/V plugin-object as defined in subclause 5.7.1 of [101] in its CE-HTML content to offer content of the Audio and/or A/V Media Class, then it shall support all Content Source related requirements in 7.5 that apply to Streaming Transfers.

[Attributes]

M	A	+RUISRC+	n/a	n/a	n/a	5YYRG	

NOTE: The following informative list show the resulting requirements:

7.5.4.2.1	7.5.4.2.2	7.5.4.2.3.1	7.5.4.2.3.3	7.5.4.2.3.4	7.5.4.2.3.6
7.5.4.2.7.2	7.5.4.2.12.2	7.5.4.2.13	7.5.4.2.14.1	7.5.4.2.14.2	7.5.4.2.14.3
7.5.4.2.14.4	7.5.4.2.15.1	7.5.4.2.15.2	7.5.4.2.16.1	7.5.4.2.16.2	7.5.4.2.16.3
7.5.4.2.16.4	7.5.4.3.2.1.1	7.5.4.3.2.1.2	7.5.4.3.2.1.3	7.5.4.3.2.2	7.5.4.3.2.4.1
7.5.4.3.2.4.2	7.5.4.3.2.4.3	7.5.4.3.2.4.5	7.5.4.3.2.4.7	7.5.4.3.2.4.9	7.5.4.3.2.4.10
7.5.4.3.2.5.1	7.5.4.3.2.5.2	7.5.4.3.2.5.3	7.5.4.3.2.6	7.5.4.3.2.7.1	7.5.4.3.2.7.2
7.5.4.3.2.7.3	7.5.4.3.2.7.4	7.5.4.3.2.7.5	7.5.4.3.2.7.6	7.5.4.3.2.8.1	7.5.4.3.2.8.2
7.5.4.3.2.9.1	7.5.4.3.2.9.2	7.5.4.3.2.10.1	7.5.4.3.2.10.2	7.5.4.3.2.10.4	7.5.4.3.2.10.5
7.5.4.3.2.10.6	7.5.4.3.2.11.1	7.5.4.3.2.11.4	7.5.4.3.2.12.1	7.5.4.3.2.12.2	7.5.4.3.2.12.3
7.5.4.3.2.12.4	7.5.4.3.2.12.5	7.5.4.3.2.12.6	7.5.4.3.2.12.7	7.5.4.3.2.14	7.5.4.3.2.15.1
7.5.4.3.2.15.2	7.5.4.3.2.15.4	7.5.4.3.2.15.6	7.5.4.3.2.15.7	7.5.4.3.2.15.8	7.5.4.3.2.15.9
7.5.4.3.2.15.10	7.5.4.3.2.16.1	7.5.4.3.2.16.2	7.5.4.3.2.18.1	7.5.4.3.2.18.2	7.5.4.3.2.18.3
7.5.4.3.2.19.1	7.5.4.3.2.19.2	7.5.4.3.2.19.3	7.5.4.3.2.19.4	7.5.4.3.2.19.5	7.5.4.3.2.19.7
7.5.4.3.2.19.8	7.5.4.3.2.19.9	7.5.4.3.2.20.1	7.5.4.3.2.20.4	7.5.4.3.2.20.5	7.5.4.3.2.20.6
7.5.4.3.2.20.7	7.5.4.3.2.20.8	7.5.4.3.2.20.9	7.5.4.3.2.20.11	7.5.4.3.2.20.12	7.5.4.3.2.20.13
7.5.4.3.2.20.14	7.5.4.3.2.20.15	7.5.4.3.2.20.17	7.5.4.3.2.21.1	7.5.4.3.2.21.2	7.5.4.3.2.22.1

7.5.4.3.2.22.2	7.5.4.3.2.22.3	7.5.4.3.2.22.4	7.5.4.3.2.22.5	7.5.4.3.2.22.6	7.5.4.3.2.22.7
7.5.4.3.2.22.8	7.5.4.3.2.22.9	7.5.4.3.2.22.10	7.5.4.3.2.22.1	7.5.4.3.2.24.1	7.5.4.3.2.24.2
7.5.4.3.2.24.3	7.5.4.3.2.24.4	7.5.4.3.2.24.5	7.5.4.3.2.24.6	7.5.4.3.2.24.7	7.5.4.3.2.24.8
7.5.4.3.2.24.9	7.5.4.3.2.24.12	7.5.4.3.2.24.13	7.5.4.3.2.25.1	7.5.4.3.2.25.2	7.5.4.3.2.31
7.5.4.3.2.32	7.5.4.3.2.33.1	7.5.4.3.2.33.2	7.5.4.3.2.33.3	7.5.4.3.2.33.4	7.5.4.3.2.34.1
7.5.4.3.2.34.2	7.5.4.3.2.34.3	7.5.4.3.2.35.1	7.5.4.3.2.35.2	7.5.4.3.2.35.3	7.5.4.3.2.35.4
7.5.4.3.2.38.1	7.5.4.3.3.1.2	7.5.4.3.3.6.3	7.5.4.3.3.15.2	7.5.4.3.3.16.1	7.5.4.3.3.16.2
7.5.4.3.3.16.3	7.5.4.3.3.16.4	7.5.4.3.3.16.5	7.5.4.3.3.16.6	7.5.4.3.3.16.9	7.5.4.3.3.19.1
7.5.4.3.3.19.2	7.5.4.3.3.20.2	7.5.4.3.3.20.3	7.5.4.3.3.20.4	7.5.4.3.3.20.5	7.5.4.3.3.20.6
7.5.4.3.3.20.7	7.5.4.3.3.21.2				

If the RUI Source capability supports RTP streaming (i.e. if one or more <audio_profile> or <video_profile> elements in the capability profile has an attribute "transport" with case-insensitive value "dlna_rtp"), then this also includes all requirements related to an RTP Serving Endpoint as indicated by 7.5.4.4.2.1 and onwards.

Note that all mandatory requirements remain mandatory, all recommendations remain recommendations and all optional requirements remain optional unless specified otherwise in this subclause (i.e. 7.8.2.6 "Combining RUI with A/V").

~~14.3.2.6.5.4~~7.8.2.6.5.4

[GUIDELINE] If a RUI Source capability uses the A/V plugin-object as defined in subclause 5.7.1 of [101] in its CE-HTML content to offer content of the Audio and/or A/V Media Class, then the RUI Source should only make use of properties and methods offered by the A/V plugin-object to control the media transport.

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	ZJXOG	

NOTE: This guideline ensures that receiving endpoints do not end up in ambiguous A/V playback states. A RUI Source cannot assume correct behavior beyond what is defined by the properties and methods of the A/V plugin object, and the requirements within this subclause of the DLNA guidelines.

~~14.3.2.6.5.5~~7.8.2.6.5.5

[GUIDELINE] If a RUI Pull Controller or a RUI Sink capability implements the A/V plugin-object (as defined in subclause 5.7.1 of [101]) of which the play()-method (as defined in [Req. 5.7.1.f] of [101]) is called with a value of 1 for argument "playspeed", then:.

- If the transport is HTTP, then the RUI Pull Controller and RUI Sink capability shall operate the same as the Play media operation in 7.5.4.3.3.2.
- If the transport is RTP, then the RUI Pull Controller and RUI Sink capability shall operate the same as the Play media operation in 7.5.4.4.6.2.36.

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	WVPGC	

~~14.3.2.6.5.6~~7.8.2.6.5.6

[GUIDELINE] If a RUI Pull Controller or RUI Sink capability is co-located with a Rendering Endpoint with an MSCP, under the conditions of 7.8.2.6.5.5, the MSCP of that device may call methods GetProtocolInfo and PrepareForConnection (if these methods are available on the server that hosts the A/V), before any of the above defined Play media operations is invoked. The device shall call these methods in accordance with 7.4.1.2.1.2.

[ATTRIBUTES]

O	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	6YQ96	

~~14.3.2.6.5.7~~7.8.2.6.5.7

[GUIDELINE] If a RUI Pull Controller or a RUI Sink capability implements the A/V plugin-object (as defined in subclause 5.7.1 of [101]) of which the stop()-method (as defined in [Req. 5.7.1.f] of [101]) is called, then:

- If the transport is HTTP, then the RUI Pull Controller and RUI Sink capability shall operate the same as the Stop media operation in 7.5.4.3.3.3.1.
- If the transport is RTP, then the RUI Pull Controller and RUI Sink capability shall operate the same as the Stop media operation in 7.5.4.4.6.2.55.

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	YYRG3	N

~~14.3.2.6.5.8~~7.8.2.6.5.8

[GUIDELINE] If a RUI Pull Controller or RUI Sink capability is co-located with a Rendering Endpoint with an MSCP, under the same conditions of 7.8.2.6.5.7, the MSCP of that device may call method ConnectionComplete (if this method is available on the server that hosts the A/V), instead of any of the above defined Stop media operations (see related 7.4.1.5.1.10, 7.4.1.5.4.1, and 7.4.1.5.4.2). The device shall call this method in accordance with 7.4.1.2.1.2).

[ATTRIBUTES]

O	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	JXOG3	

~~14.3.2.6.5.9~~7.8.2.6.5.9

[GUIDELINE] If a RUI Pull Controller or a RUI Sink capability implements the A/V plugin-object (as defined in subclause 5.7.1 of [101]) of which the play()-method (as defined in [Req. 5.7.1.f] of [101]) is called with a value not equal to 0 or 1 for argument "playspeed", then the RUI Pull Controller and RUI Sink capability shall operate according to the fast and slow scan operation guidelines for forward and backward playback as follows:

- If the transport is HTTP, then the RUI Pull Controller and RUI Sink capability shall at least operate according to 7.5.4.3.3.8.1, 7.5.4.3.3.9.1, 7.5.4.3.3.10.1, 7.5.4.3.3.11.1, 7.5.4.3.3.12.1, 7.5.4.3.3.15.1, 7.5.4.3.3.16.3, 7.5.4.3.3.16.8, 7.5.4.3.3.21.1, and 7.5.4.3.2.10.7.
- If the transport is RTP, then the RUI Pull Controller and RUI Sink capability shall operate the same as the SCAN media operation defined in 7.5.4.4.6.2.43.

In addition, the following requirements shall be met:

- The playspeed values within the media transport operation request to the server, shall be rounded to the nearest fractional value as indicated of being supported through the sp-param of the 4th field protocolInfo value.
- After invocation to the play() method, the resulting playspeed shall be made available through the "speed" property of the A/V plugin object.

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	VPGCA	

~~14.3.2.6.5.10~~7.8.2.6.5.10

[GUIDELINE] If a RUI Pull Controller or a RUI Sink capability implements the A/V plugin-object (as defined in subclause 5.7.1 of [101]) of which the play()-method (as defined in [Req.

5.7.1.f] of [101]) is called with a value of 0 for argument "playspeed", then it shall operate according to the media operation guidelines for pause as follows:

- If the transport is HTTP, then the RUI Pull Controller and RUI Sink capability shall operate the same as the Pause media operation defined in 7.5.4.3.3.4 through 7.5.4.3.3.6.
- If the transport is RTP, then the RUI Pull Controller and RUI Sink capability shall at least operate the same as the Pause media operation defined in 7.5.4.4.6.2.47 through 7.5.4.4.6.2.50.

In addition, the following requirements shall be met:

- Any of the intermediate media operations resulting from the use of the MT pause/release method shall not trigger a state change in the A/V plugin-object's playState variable, i.e. if the media continues playing after a pause, only a state transition from playstate value 2 to playstate value 1 shall be triggered.
- related to the Pause media operation.

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	YQ96L	

~~14.3.2.6.5.11~~7.8.2.6.5.11

[GUIDELINE] If a RUI Pull Controller or a RUI Sink capability implements the A/V plugin-object (as defined in subclause 5.7.1 of [101]A) of which the seek()-method (as defined in [Req. 5.7.1.f] of [101]) is called, then it shall operate according to the seek media operation as follows:

- If the transport is HTTP, then the RUI Pull Controller and RUI Sink capability shall operate the same as the Seek media operation in 7.5.4.3.3.7.1.
- If the transport is RTP, then the RUI Pull Controller and RUI Sink capability shall operate the same as the Seek media operation defined in 7.5.4.4.6.2.39 through 7.5.4.4.6.2.41.

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	YRG3W	

NOTE: For seeking through protected content, additional requirements apply. See 7.8.2.6.7.1 for more information.

~~14.3.2.6.5.12~~7.8.2.6.5.12

[GUIDELINE] If a RUI Pull Controller or RUI Sink capability is co-located with a Rendering Endpoint with an MSCP, under the same conditions of 7.8.2.6.5.9, 7.8.2.6.5.10, or 7.8.2.6.5.11, the MSCP of that device may call method GetProtocolInfo (if this method is available on the server that hosts the A/V), before any of the media operations defined in the corresponding requirement is invoked. The device shall call this method in accordance with 7.4.1.2.1.2.

[ATTRIBUTES]

O	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	XOG33	

~~14.3.2.6.5.13~~7.8.2.6.5.13

[GUIDELINE] If the device to which the RUI Pull Controller or RUI Sink capability is added is a UPnP A/V MediaRenderer, and if the capability that implements the A/V plugin-object (as defined in subclause 5.7.1 of [101]) supports the methods getTransportStateVariables() and setTransportStateVariables() (as defined in [Req. 5.7.1.f] of [101]), then the variables returned by the methods shall be consistent with the state variables of the AVTransport service.

[Attributes]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	PGCA4	

~~14.3.2.6.6~~7.8.2.6.6 Content Protection Guidelines

~~14.3.2.6.6.1~~7.8.2.6.6.1

[Guideline] If a RUI Source capability exposes a res@protocolInfo through the <param>-element "dlna-res-attr" for the A/V plugin-object as defined in subclause 5.7.1 of [101] in its CE-HTML content, bit 16 of the 4th field of ProtocolInfo shall denote LP-flag (Link-Protected Content Flag), and all guidelines associated with this flag shall be adhered to. Specifically:

- LP Flag applies to all DLNA transport protocols.
- If the flags-param is omitted then this flag shall have an inferred value of false.
- If the cleartextbyteseek-full flag or the lop-cleartextbytes-flag are set then this flag shall be set to true. See the following for more information: 7.3.6 MM CP: LP-flag (Link Protection Flag) in [57].
- If the content binary described by the protocolInfo uses DLNA Link Protection when it is transmitted then the LP-flag shall be true.

If the content binary described by the protocolInfo does not use DLNA Link Protection when it is transmitted then the LP-flag shall be false if present.

[Attributes]

M	A	+RUISRC+	n/a	n/a	n/a	Q96L7	

NOTE: A/V embedded inside Remote User Interfaces have to abide by content protection guidelines for protected content. See also 7.8.2.6.4.4 about the related getContentFeatures.dlna.org HTTP header.

~~14.3.2.6.6.2~~7.8.2.6.6.2

[Guideline] If the "data"-attribute of the A/V plugin-object as defined in subclause 5.7.1 of [101] inside the CE-HTML content, specifies a content binary that uses DLNA Link Protection, then all applicable DLNA Link Protection guidelines shall be met. Please note that Link Protection can be triggered via either of the following two mechanisms:

- LP flag of the 4th field of ProtocolInfo (as defined through the <param>-element "dlna_res_attr" or getContentFeatures.dlna.org HTTP header related to the A/V object) is set.
- The content binary uses a DLNA media profile ID denoting protected content (e.g. DTCP_, WMDRM_, etc) in the DLNA.ORG_PN parameter of the 4th field of ProtocolInfo.

[Attributes]

M	A	+RUISRC+ +RUISINK+ +RUIPL+	n/a	n/a	n/a	RG3WV	

NOTE: At present, DTCP/IP and WM DRM are the only two approved content protection protocols. However, in future, other DLNA media profile IDs might be defined for other content protection protocols.

~~14.3.2.6.6.3~~7.8.2.6.6.3

[Guideline] If a RUI Pull Controller or a RUI Sink capability implements the A/V plugin-object (as defined in subclause 5.7.1 of [101]) of which the seek()-method (as defined in [Req. 5.7.1.f] of [101]) is called for protected content, then the following guidelines for the seek media operation as defined in DLNA Link Protection Guidelines document are also applicable:

- If the transport of the protected content is HTTP, then the RUI Pull Controller and RUI Sink capability shall act the same as the Seek media operation in 7.6.4.3.2, 7.6.4.3.4, and 7.6.4.4.2 of [57]. When using DTCP-IP link protection system, 8.6.1.2 and 8.6.1.3 of [57] are also applicable. When using WM-DRM link protection system, guidelines 9.5.1.7 and 9.5.1.8 of [57] are also applicable.
- If the transport of the protected content is RTP, then guideline 7.6.4.3.2 of [57] is applicable.

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	[57]	OG33U	

NOTE: This guideline is applicable for seek operation on protected content only. Other guidelines referred to in this guideline (e.g. 7.4.12, etc.) can be found in the DLNA Link Protection Guidelines document. For more information on DLNA Seek Models for protected content, please see A.3 in [57]. For more information on Seek Operations on protected content see A.6 in [57].

~~14.3.2.6.7~~7.8.2.6.7 Other Guidelines related to binding RUI with A/V

~~14.3.2.6.7.1~~7.8.2.6.7.1

[GUIDELINE] If a RUI Source capability implements playlist functionality through the remote UI to playback a (possibly repeated) sequence of audio and/or video content items by a RUI Pull controller or RUI Sink capability, then the RUI Source should include Javascript that monitors the "playState" property of the A/V plugin object (as defined in subclause 5.7.1 of [101]). Once the playState becomes "5" then the Javascript should change the "data" attribute of the A/V plugin object with the URL of the next content item to play, and then the play() method can be called to start playing this next A/V content item.

[ATTRIBUTES]

S	A	+RUISRC+	n/a	n/a	n/a	GCA4B	

NOTE: This is the recommended way to implement playlist functionality through remote UI, since this method is compatible with all RUI Pull controllers and RUI Sink capabilities that support A/V playback. This does not preclude the RUI Source from supporting a playlist container format, such as DIDL_S (as defined in [56], 11.1). See 7.8.2.6.7.2 for more information.

~~14.3.2.6.7.2~~7.8.2.6.7.2

[GUIDELINE] If a RUI Pull Controller or RUI Sink capability has indicated support for a playlist container format in its capability description (as defined by [Req. 5.2.1.a] and [Req. 5.2.1.c] of [101]), then a RUI Source may use the playlist container format for the content referred to by the "data" attribute of the A/V plugin object. In this case, the RUI source may use the "nrTracks", "currentTrackIndex" properties and the "next" and "previous" methods of the A/V plugin object to control the playback of the items in the playlist.

[ATTRIBUTES]

O	A	+RUISRC+	n/a	n/a	n/a	96L7G	

NOTE: This allows a RUI Source to support a playlist container format, such as DIDL_S (as defined in [56], 11.1) for offering playlist functionality.

~~14.3.2.7~~7.8.2.7 Presentation and Control

~~14.3.2.7.1~~7.8.2.7.1 General

This subclause describes the guidelines related to presentation and control of CE-HTML content. It has two subclauses. The first subclause (called "General Presentation and Control guidelines") contains the guidelines related to the exact message and content format of CE-HTML pages over the network and how they should be interpreted to prevent browser ambiguities. The second subclause (called "General browser implementation and UI authoring suggestions") contains some general browser implementation and UI authoring suggestions that provide clarification and guidance to implementers and UI authors.

~~14.3.2.7.2~~7.8.2.7.2 General Presentation and Control Guidelines

~~14.3.2.7.2.1~~7.8.2.7.2.1

[GUIDELINE] The RUI Pull Controller, the RUI Sink capability and the RUI Source capability shall adhere to the CE-HTML content requirements as defined in subclause 5.4 of [101].

[ATTRIBUTES]

M	A	+RUISRC+ +RUISINK+ +RUIPL+	n/a	n/a	n/a	G3WVZ	

~~14.3.2.7.2.2~~7.8.2.7.2.2

[GUIDELINE] The RUI Pull Controller, the RUI Sink capability and the RUI Source capability shall adhere to the detailed CE-HTML implementation and authoring requirements as defined in Annexes G through I of [101].

[ATTRIBUTES]

M	A	+RUISRC+ +RUISINK+ +RUIPL+	n/a	n/a	n/a	G33U5	

~~14.3.2.7.2.3~~7.8.2.7.2.3

[GUIDELINE] The RUI Pull Controller, the RUI Sink capability and the RUI Source capability shall adhere to the cookie guidelines as defined in subclause 5.9 and the robustness guidelines in subclause 5.10 of [101].

[ATTRIBUTES]

M	A	+RUISRC+ +RUISINK+ +RUIPL+	n/a	n/a	n/a	CA4B7	

~~14.3.2.7.2.4~~7.8.2.7.2.4

[GUIDELINE] The RUI Pull controller and the RUI Sink capability shall support [Req. 5.2.3.a] through [Req. 5.2.3.f] of [101] regarding the browser area.

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	6L7GA	

NOTE: The browser area is the actual display area on the screen that is available for the browser to render the CE-HTML content. Note that the browser area does not include virtual screen estate available through scrolling. Note also the general implementation and authoring guidelines 5) and 6) in the subclause "General browser implementation and UI authoring suggestion below, about dealing with overscan.

~~14.3.2.7.2.5~~7.8.2.7.2.5

[GUIDELINE] If a RUI Pull controller or a RUI Sink capability has indicated support for A/V playback by including a <video_profile>-element in its capability description (as defined in 7.8.2.3), and the value of the <overlay>-element in the capability profile is unequal to "none", the size of the Browser Area of a RUI Pull Controller and RUI Sink capability shall exactly match the <width> and <height> values as indicated in the capability profile of the RUI Pull controller or RUI Sink capability. The same holds if the RUI Pull Controller or RUI Sink capability has indicated support for the "video/local" plugin (as defined in [101], 5.7.2) through a <mime-extension>-element, and the value of the <overlaylocal>-element is unequal to "none".

[Attributes]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	3WVZV	

NOTE: This requirement relates to the precise positioning of overlays of CE-HTML content on top of video. In order to achieve this, it is important that the actual size of the browser area does not deviate from what the RUI Pull Controller or RUI Sink capability has specified during the capability exchange (i.e. does not use any predefined offset and does not have to use scrolling for rendering the content inside the browser area, if that content exactly matches the size as indicated through the capability profile).

~~14.3.2.7.2.6~~7.8.2.7.2.6

[Guideline] If a RUI Pull controller or a RUI Sink capability has indicated support for A/V playback by including a <video_profile>-element in its capability description (as defined in7.8.2.3), and the value of the <overlay>-element in the capability profile is unequal to "none", a RUI Pull controller and RUI Sink capability shall ensure that the browser area covers exactly the same area on the display as the full screen rendering of video content. The same holds if the RUI Pull Controller or RUI Sink capability has indicated support for the "video/local" plugin (as defined in subclause 5.7.2 of [101]) through a <mime-extension>-element, and the value of the <overlaylocal>-element is unequal to "none".

[Attributes]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	33U5B	

NOTE: This requirement relates to the precise positioning of overlays of CE-HTML content on top of video. In order to achieve it is important that the location and the actual display size of full screen video matches with the location and the actual display size of the full screen browser area. If the aspect ratio of the video changes, the browser area does not change (see related 7.8.2.7.2.8).

~~14.3.2.7.2.7~~7.8.2.7.2.7

[Guideline] If a RUI Pull controller or a RUI Sink capability has indicated support for A/V playback by including a <video_profile>-element in its capability description (as defined in7.8.2.3), and the value of the <overlay>-element in the capability profile is unequal to "none", a RUI Pull controller and RUI Sink capability shall ensure that the aspect ratio of the video remains the same as indicated in the value of the <param>-element with the name "aspectratio" of the A/V object as defined in [101], 5.7.1, if such <param> is included and ends with the case-insensitive string ";keep=true" or "; keep=true". If the <param>-element with name "aspectratio" is not included or ends with ";keep=false", "; keep=false", or does not include the string "keep", the RUI Pull Controller and RUI Sink capability may change the aspect ratio of the video that is shown.

[Attributes]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	A4B73	

NOTE: This requirement relates to the precise positioning of overlays of CE-HTML content on top of video. In order to achieve it is important to know that the aspect ratio of the video either stays the same or not. Here is an example:

```
<object type="video/mpeg" id="vid1"
    data="http_//www.avsource.com/video/test.mpg">
    <param name="dlna_res_attr"
    value="protocolInfo=&quot;http-
    get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC&quot;"/>
    <param name="aspectratio" value="1.33;keep=true"/>
        DLNA.ORG_PN=MPEG_PS_NTSC&quot;"/>
</object>
```

~~14.3.2.7.2.8~~7.8.2.7.2.8

[GUIDELINE] If a RUI Pull controller or a RUI Sink capability has indicated support for A/V playback by including a <video_profile>-element in its capability description (as defined in7.8.2.3), and the value of the <overlay>-element in the capability profile is unequal to "none", a RUI Pull controller and RUI Sink capability shall support the following event handler properties on the A/V object in addition to the properties and methods as defined in subclause 5.7.1 of [101]:.

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	L7GAM	

NOTE: This requirement relates to the precise positioning of overlays of CE-HTML content on top of video. In order to achieve it is important to know that the aspect ratio of the video either stays the same (see 7.8.2.7.2.7) or is made available to the CE-HTML page author.

~~14.3.2.7.2.9~~7.8.2.7.2.9

[GUIDELINE] If a RUI Source capability uses pixel precise positioning (i.e. uses "px" units for CSS-attributes related to the <body>-element or direct descendants of the <body>-element) inside the CE-HTML page, it shall append the width and height parameters to the "application/ce-html+xml" value for the HTTP-header "Content-Type" separated by semicolons as defined by bullets 8 b) through 8 e) of [Req. 5.3.a] of [101]. In addition, it shall append a boolean parameter called "relativepos" with value "false" after the width and height parameters, separated by a semicolon.

[ATTRIBUTES]

M	A	+RUISRC+	n/a	n/a	n/a	WVZV8	

NOTE: The following example shows a valid HTTP Content-Type header in case pixel precise positioning is used:

```
Content-Type: application/ce-html+xml;charset="utf-8";width="320";
    height="240;relativepos="false"
```

A page author is free to provide these parameters (i.e. width, height, together with 'relativepos="false"'), also in any other case in which he/she finds it important that the client respects the width and height of the page as indicated through parameters "width" and "height" (see 7.8.2.7.2.10 and 7.8.2.7.2.11 for more information about rendering).

Please note that pixel precise positioning is discoraged, in particular for content designed for the MD_UIPROF profile, to cope with the large variation in screen sizes of mobile clients (i.e. M-DMP or other M-* devices with a RUI Pull Controller or RUI Sink capability) that the MD_UIPROF profile tries to encompass.

~~14.3.2.7.2.10~~7.8.2.7.2.10

[GUIDELINE] If a RUI Source capability uses pixel precise positioning (i.e. uses "px" units for CSS-attributes related to the <body>-element or direct descendants of the <body>-element) inside the CE-HTML page, then in addition to the width, height and relativepos-parameters as defined in 7.8.2.7.2.9, it may append the following name-value pairs:

alignX=("left"|"right"|"center"|int), and alignY=("top"|"bottom"|"center"|int)

The semantics of the values are defined as in 7.8.2.7.2.11. If included, the parameters shall be separated with semicolons and shall be appended right after attribute "relativepos" in the same order in which they are defined above.

[ATTRIBUTES]

O	A	+RUISRC+	n/a	n/a	n/a	3U5BV	

NOTE: The following example shows a valid HTTP Content-Type header in case pixel precise positioning is used with an indication of where the content can be placed inside the browser area (see 7.8.2.7.2.5 for more information):

Content-Type: application/ce-html+xml;charset="utf-8";width="320";
height="240;relativepos="false";alignX="right"; alignY="bottom"

Please note that pixel precise positioning is discouraged, in particular for content designed for the MD_UIPROF profile, to cope with the large variation in screen sizes of mobile clients (i.e. M-DMP or other M-* devices with a RUI Pull Controller or RUI Sink capability) that the MD_UIPROF profile tries to encompass.

~~14.3.2.7.2.11~~7.8.2.7.2.11

[GUIDELINE] If a RUI Pull controller or RUI sink capability encounters parameters "width", "height", "relativepos", "alignX" and/or "alignY", appended to "application/ce-html+xml" in the HTTP Header "Content-Type", it shall render the CE-HTML page as follows:

a) If it contains parameters "width", "height", parameter "relativepos" with value "false", and parameters "alignX" and "alignY", the following semantics hold for the values of the "alignX"-parameter defined below:

- "left": means that the content shall be rendered with a horizontal offset of 0 pixels, if rendered in a browser area that is wider than indicated through the "width" parameter in the Content-Type HTTP header.
- "right": means that the content shall be rendered with a horizontal offset of (width of browser area minus the value of the "width" parameter in the Content-Type HTTP header), if rendered in a browser area that is wider than indicated through the "width" parameter in the Content-Type HTTP header.
- "center": means that the content shall be rendered with a horizontal offset of (width of browser area minus the value of the "width" parameter in the Content-Type HTTP header) / 2, if rendered in a browser area that is wider than indicated through the "width" parameter in the Content-Type HTTP header.
- Any integer value greater than or equal to zero: this means that the content shall be rendered with a horizontal offset (in pixels) as indicated by the integer value, if rendered in a browser area that is wider than indicated through the "width" parameter in the Content-Type HTTP header. If the integer value is bigger than (width of browser area minus the value of the "width" parameter in the Content-Type HTTP header), then it shall be handled equivalent to alignX="right".

For parameter "alignY", the following semantics hold:

- "top": means that the content shall be rendered with a vertical offset of 0 pixels, if rendered in a browser area that is higher than indicated through the "height" parameter in the Content-Type HTTP header.
- "bottom": means that the content shall be rendered with a vertical offset of (height of browser area minus the value of the "height" parameter in the Content-Type HTTP header), if rendered in a browser area that is higher than indicated through the "height" parameter in the Content-Type HTTP header.
- "center": means that the content shall be rendered with a vertical offset of (height of browser area minus the value of the "height" parameter in the Content-Type HTTP header) / 2, if rendered in a browser area that is higher than indicated through the "height" parameter in the Content-Type HTTP header.
- Any integer value greater than or equal to zero: this means that the content shall be rendered with a vertical offset (in pixels) as indicated by the integer value, if rendered in a browser area that is higher than indicated through the "height" parameter in the Content-Type HTTP header. If the integer value is bigger than (height of browser area minus the value of the "height" parameter in the Content-Type HTTP header), then it shall be handled equivalent to alignY="bottom".

If the browser area is not wider in case of "alignX" or higher in case of "alignY", the parameter values for "alignX" respectively "alignY" should be ignored.

b) If it contains parameters "width", "height", and parameter "relativepos" with value "false", without parameters "alignX" and "alignY", it shall render the CE-HTML page the same as defined under bullet 1 for 'alignX="left";alignY="top"'.

c) If it contains no parameters or it contains parameters "width", "height", without parameter "relativepos" or it contains parameters "width", "height" and parameter "relativepos" with value "true", the RUI Pull controller and RUI Sink capability may assume that the CE-HTML page does not use absolute positioning and hence may draw the page using the full width and height of the browser area, without any particular alignment/offset as defined under bullets 1 and 2.

[ATTRIBUTES]

M	A	+RUISRC+ +RUIPL+	n/a	n/a	n/a	4B73O	

NOTE: Note that for rendering the content with a given offset inside the browser area, the coordinate system for the author must not change, i.e. coordinate (0,0) will just not correspond anymore to the top-left corner of the browser area, but a place somewhere inside the browser area, and (width-1, height-1) will not just correspond anymore to the bottom-right corner of the browser area (or a virtual part of the browser area that can only be reached by scrolling. Note that scrolling must adhere to the requirements as defined by [Req. 5.2.3.b], [Req. 5.2.1.a], bullet 8c of [Req. 5.3.a], and bullet 2 of the general authoring requirements in Annex H of [101], and 7.8.2.7.2.4 and 7.8.2.7.2.5 as defined above.

This requirement allows for variations in screen sizes, while still maintaining control over where on the screen the absolutely positioned content will be drawn. In other words, authors do not specially have to author for each possible variation in display size as indicated through the <width> and <height> attributes. Instead authors can just author for one or more of the predefined UI profiles (as defined by [Req. 5.2.1.b] of [101]) or popular screen sizes, and use the "alignX" and "alignY" attributes as a predictable way of placing the UI on the screen, in particular for the precise positioning of overlays of CE-HTML content on top of video.

Please note that pixel precise positioning is discouraged, in particular for content designed for the MD_UIPROF profile, to cope with the large variation in screen sizes of mobile clients (i.e. M-DMP or other M-* devices with a RUI Pull Controller or RUI Sink capability) that the MD_UIPROF profile tries to encompass.

~~14.3.2.7.2.12~~7.8.2.7.2.12

[GUIDELINE] The RUI Pull Controller and the RUI Sink capability shall support CSS properties "nav-up", "nav-down", "nav-left", and "nav-right" as defined in subclause 10.2.2 of CSS3 Basic User Interface Module (http://www.w3.org/TR/css3-ui/#nav-dir). These properties shall adhere to [Req. 5.4.1.f] of [101] for navigation keys. .

[ATTRIBUTES]

M	A	+RUISINK+ +RUIPL+	n/a	n/a	n/a	7GAMU	

NOTE: This relates to bullet 4 of the general implementation and authoring suggestions in the "General browser implementation and UI authoring suggestions" subclause below.

~~14.3.2.7.3~~7.8.2.7.3 General browser implementation and UI authoring suggestions (informative)

Although the following general browser implementation and UI authoring suggestions are not required for interoperability, they provide clarification and guidance to browser implementers and CE-HTML content authors. This list is not complete. We leave it to the device manufacturer that includes a CE-HTML browser, and to the CE-HTML content author to make the right choices here:

a) A CE-HTML browser should show a visual indication when CE-HTML content is being loaded, and should provide helpful error messages in case a connection fails or times out.

b) A content author should provide an indication of the progress (e.g. through CE-HTML's Notifsocket mechanism) inside the CE-HTML page, once a connection request by a RUI pull controller or RUI sink capability is successfully received and the server is expected to take a considerable amount of time (e.g. 10 seconds) to respond. This is a general UI authoring guideline that applies to networked applications (see http://www.useit.com/papers/responsetime.html, and Ben Schneiderman's book "Designing the User Interface: Strategies for Effective Human-Computer Interaction" for more information).

c) To prevent unnecessary delays, a RUI Pull Controller or RUI Sink capability that was showing a full-screen A/V stream coming from the network before switching to the CE-HTML browser to show some CE-HTML content, may free up some networking (and processing) resources by pausing the current A/V stream before connecting to a RUI Source. This might improve the responsiveness of the UI, since RUI messages have a lower QoS level than A/V (see 7.8.2.8.1).

d) A CE-HTML content author should avoid heavy usage of Javascript in its CE-HTML content, because of the wide variety of processing power client devices have that implement the DLNA Remote UI guidelines. This can/should be done not only for focus navigation (by using the "nav-up", "nav-down", … CSS attributes, see 7.8.2.7.2.12), but for example also by restricting the usage of timers (in particular timers with timeout periods smaller than 1 second), reduce string sizes and reduce the number of dynamic style changes through DOM Level 2 style.

e) A CE-HTML browser should not obscure parts of the browser area due to overscan.

f) A CE-HTML content author should take into account that some devices might still obscure parts of the browser area due to overscan. Therefore, he/she should author the content in such a way that no form elements or important data items are shown within 5% of the edges of the browser area.

~~14.3.2.8~~7.8.2.8 Quality of Service

~~14.3.2.8.1~~7.8.2.8.1

[GUIDELINE] If DLNAQOS as defined in 7.2 is implemented, Remote User Interface protocol messages shall be tagged with DLNAQOS_1, or a lower DLNAQOS_UP value (where "or a lower" is defined by 7.2.4.2.2.2 and 7.2.4.2.2.3, for both requests and responses in accordance with Table 11. Remote User Interface protocol messages include HTTP requests and responses that are the result of a user input action on the remote UI, messages that are sent through a TCP connection created through a NotifSocket plugin (as defined in subclause 5.5.1 of [101]), third party notifications, and any other traffic defined in subclauses 5.2.2 through 5.6 and subclauses 5.8 and 5.9 of [101].

[ATTRIBUTES]

M	A	+RUISRC+ +RUISINK+ +RUIPL+	n/a	n/a	n/a	VZV8W

NOTE: In addition to Remote User Interface protocol messages there are two other types of traffic related to remote UI, each with their own DLNAQOS_UP value:

- For A/V content that is combined with remote UI: see the requirements defined in 7.5.4.2.12 of the "Media Transport" guidelines (referenced by 7.8.2.6.5.2 and 7.8.2.6.5.3 of the "Combining RUI with A/V" guidelines), and item 3 of the general implementation and authoring guidelines in the "General browser implementation and UI authoring suggestions" subclause at the end of 7.8.2.7 "Presentation and Control".
- For UPnP Device and control point traffic for remote UI (as defined in7.8.2.1): see 7.3.2.36.

Annex A
(informative)

Network Infrastructure Device (NID) Recommendations

A.1 General

Network Infrastructure Devices (NID) are outside the scope of the DLNA Home Networked Device Interoperability Guidelines. However, since DLNA devices interact with each other on a home network, that network and its infrastructure greatly influence the user experience. Network Infrastructure Devices that abide by the recommendations in this subclause will contribute to and facilitate interoperability and a good user experience with DLNA devices. Although this document lists recommendations, a NID can not be said to conform to this annex unless it implements all the items that apply to it marked with the 'S' compliance classifier.

A.2 NID Functions

The recommendations in Table A.2 refer to different types of NID functionality. A NID can be a single function device, such as a switch, or it can be a combination device that implements multiple functions such as a wireless access point that also provides Ethernet ports with bridging between wired and wireless interfaces. The NID functions referenced in the recommendations are defined in Table A.1.

Table A.1 — NID Functions

Device Function	Descriptions
Access Point (AP)	APs are 802.11 hubs, the central points of contact in 802.11 wireless networks. APs typically include bridges (see Bridge below) between 802.11 and 802.3 network segments.
Bridge	Bridges connect two networks of different physical media types with translation between formats of the media types occurring at layer 2 of the ISO model.
Interconnect	Interconnects are device functions such as switches or hubs that connect two network segments of the same type (e.g. Ethenet segments). This annex recommends that all interconnects be switches. Hub functionality should be avoided on the home network.
Internet Gateway Device (IGD)	IGDs interface the home network to the public Internet. IGDs present different interfaces with different characteristics to their LAN side-the home network-and their WAN side-the public Internet.
Router	Routers pass traffic between two or more IP subnets and, within a single subnet, perform address resolution of IP addresses. Routers can be considered to do translation between networks at layer 3 of the ISO model.
Switch	Switches route network traffic by MAC address, layer 2 of the ISO model, within a single subnet.

A.3 NID Recommendations

A.3.1 General Capability Recommendations: Ethernet

A.3.1.1 NC NID Ethernet: Base

[GUIDELINE] If Ethernet is supported, IEEE 802.3i (10BASE-T) and 802.3u (100BASE TX) with auto negotiation capability and a connection to the network provided by an RJ45 connector is recommended.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[3]	POHR6	

A.3.1.2 NC NID Ethernet: Cabling

[GUIDELINE] If Ethernet is supported, any supplied network cabling should have a rating of Category 5e or better.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[87]	KHP8F	

A.3.1.3 NC NID Ethernet: Gigabit

[GUIDELINE] If Ethernet is supported, IEEE 802.3ab (1000BASE T) is optionally recommended in addition to A.2.1. An implementation should support auto negotiation of gigabit operation with a similarly capable link partner and drop down to a lower speed as appropriate.

[ATTRIBUTES]

O	R	n/a	n/a	n/a	[3]	OHR6E	

A.3.1.4 NC NID Ethernet: QoS Tolerance

[GUIDELINE] If Ethernet is supported, tagged packets should be tolerated. Tagged packets are Ethernet packets that include priority tags conformant with [3], subclause 3.5 entitled 'Elements of the Tagged MAC Frame'. Here, 'tolerate' means passing the packet, including the packet tag, without alteration, and without appreciable performance penalty. In cases where a tagged packet is passed to a higher network layer, the packet payload should be passed up identically to the way it would be if the packet were not tagged. Devices may also honor the priority indication in a packet tag, passing the packet in priority order with respect to other packets in the traffic load.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[1][3]	XKHP8	

A.3.2 Device Recommendations: IGD

A.3.2.1 NC NID IGD: LAN Side IP Stack

[GUIDELINE] On their LAN side interface, IGDs should support a TCP/IP stack that includes IPv4, TCP, UDP, ARP, and ICMP components conformant to all required protocol aspects defined in [16] and [20].

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[11] [12] [13] [14] [15] [16] [20]	9XKHP	

A.3.2.2 NC NID IGD: LAN Side DHCP

[GUIDELINE] On their LAN side interface, IGDs should support a DHCP service that provides home network clients with an IP address, a subnet mask, a DNS server address, and a default gateway address. On power up, the DHCP server should send a network advertisement of DHCP service.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[22]	SGKWD	

A.3.2.3 NC NID IGD: LAN Side DNS

[GUIDELINE] On their LAN interface, IGDs should support a DNS service capable of resolving DNS references or allow pass through of DNS requests to an external DNS server.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[73]	HP8F3	

A.3.2.4 NC NID IGD: NAT

[GUIDELINE] IGDs should support NAT functionality between their LAN side and WAN side interfaces.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[72]	7TSGK	

A.3.2.5 NC NID IGD: Upgradeability

[GUIDELINE] IGDs should be firmware updatable by the end user.

[ATTRIBUTES]

S	?	n/a	n/a	n/a	n/a	TSGKW	

A.3.3 Device Recommendations: AP

A.3.3.1 NC NID AP: Connectivity

[GUIDELINE] APs should support either 802.11n or both 802.11a and 802.11g, with concurrent operation (both 2.4GHz and 5GHz clients simultaneously) and bridging between the two wireless segments. APs should include Ethernet connectivity conformant to all [NC NID Ethernet:] labeled requirements in this table with bridging between the Ethernet and 802.11 segments.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[84] [100]	IA9O6	C

NOTE: Note that 802.11g also includes support for 802.11b.

A.3.3.2 NC NID AP: Wi-Fi Conformance

[GUIDELINE] APs should conform to Wi-Fi test plan requirements at the time the product is offered to the market.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[84] [86] [100]	77T3R	C

NOTE: Wi-Fi interoperability requirements are increasing with time as new capabilities and features are specified by IEEE 802.11. When these capabilities are added to the Wi-Fi certification test plans, wireless implementations are encouraged to conform to them.

A.3.3.3 NC NID AP: Upgradeability

[GUIDELINE] APs should be firmware updatable by the end user.

[ATTRIBUTES]

S	A	n/a	n/a	n/a	n/a	T3RKJ	

A.3.3.4 NC NID AP: QoS Support

[GUIDELINE] APs should support DLNAQOS on all their network interfaces in accordance with the recommendations in guidelines A.2.14 through A.2.16.

[ATTRIBUTES]

S	A	n/a	n/a	n/a	n/a	YXIA9	

A.3.3.5 NC NID AP: 802.11 QoS Support

[GUIDELINE] If DLNAQOS is supported on an 802.11 network interface by an AP, it should be conform to all Wi-Fi WMM mandatory requirements.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[85] [86]	7T3RK	

NOTE: QoS support is optional but if supported, conformance to Wi-Fi requirements is encouraged.

WMM provides the base level QoS specification for 802.11 network devices.

A.3.3.6 NC NID AP: WMM Access Category Mapping

A.3.3.6.1

[GUIDELINE] If WMM is supported on an 802.11 network interface by an AP and it is bridging between the 802.11 network interface and an 802.3 network interface, packets received on the 802.11 network interface and transmitted on the 802.3 network interface should include the IEEE 802.1D user priority value in the IEEE 802.1Q header and the DSCP tag corresponding to the WMM Access Category of the received 802.11 packets in accordance with the following table:

Table A.2 — WMM Access Category Mapping

WMM Access Category	IEEE 802.1D priority		DSCP
AC_BK	1	BK	0x08
AC_BE	0	BE	0x00
AC_VI	5	VI	0x28
AC_VO	7	NC	0x38

[ATTRIBUTES]

S	A	n/a	n/a	n/a	[1] [2] [85] [32]	XIA9O	

NOTE: In the case of bridging 802.3 traffic onto 802.11, the WMM test plan defines mapping from DSCP and 802.1Q into WMM priorities; however Wi-Fi does not mandate which approach to implement. This yields an interoperability problem that is addressed by this guideline.

A.3.3.6.2

[GUIDELINE] If WMM is supported on an 802.11 network interface by an AP and it is bridging between the 802.11 network interface and an 802.3 network interface, packets received on the 802.3 network interface and transmitted on the 802.11 network interface should include the WMM Access Category corresponding to the IEEE 802.1D user priority value in the IEEE 802.1Q header tag of the received 802.3 packets in accordance with the following table:

Table A.3 — WMM Access and IEEE 802.1D Priority

IEEE 802.1D priority		DSCP	WMM Access Category
1	BK	0x08	AC_BK
2	-	0x10	
0	BE	0x00	AC_BE
3	EE	Ox18	
4	CL	0x20	AC_VI
5	VI	0x28	
6	VO	0x30	AC_VO
7	NC	0x38	

[ATTRIBUTES]

S	A	n/a	n/a	n/a	[1] [2] [85] [32]	GKWDR	

A.3.3.6.3

[GUIDELINE] If WMM is supported on an 802.11 network interface by an AP that is bridging between the 802.11 network interface and an 802.3 network interface and an 802.3 packet is received that does not contain an 802.1Q tag, the AP should look at the DSCP tag and map that to a WMM Access Category in accordance with the table in A.2.15.2 and preserve the DSCP tag across the 802.3 and 802.11 segments.

[ATTRIBUTES]

S	A	n/a	n/a	n/a	[1] [85] [32]	3RKJV	

A.3.3.6.4

[GUIDELINE] If an AP receives an 802.3 packet that does not contain an 802.1Q or a DSCP tag, the packet should be passed through to the 802.11 interface unmodified without the addition of any priority tagging.

[ATTRIBUTES]

S	A	n/a	n/a	n/a	[1] [85] [32]	A9O67	

A.3.3.6.5

[GUIDELINE] If an AP receives an 802.11 packet that is not tagged with a WMM Access Category, the packet should be passed through to the 802.3 interface unmodified without the addition of any priority tagging.

[ATTRIBUTES]

S	A	n/a	n/a	n/a	[1] [85] [32]	7VHRU	

A.3.3.7 NC NID AP: WMM Admission control

[GUIDELINE] An AP should not require an admission control procedure for any access category (AC) on an 802.11 network interface. The AP should advertise that admission control is not required in the ACM flags of the WMM parameter elements.

[ATTRIBUTES]

S	?	n/a	n/a	n/a	[85]	HRUVW	

NOTE: This guideline allows DLNA Device Classes with 802.11 network interfaces to properly use AC_VO and AC_VI for DLNAQOS_3 and DLNAQOS_2 respectively.

A.3.3.8 NC NID AP: Wi-Fi Simple Config Conformance

[GUIDELINE] APs should conform to clause 4 of the Wi-Fi Simple Config test plan requirements at the time the product is offered to market.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[99]	7JZ3X	C

A.3.4 Device Recommendations: Bridge
NC NID Bridge: Addressability

[GUIDELINE] All bridges should be IP addressable and have a unique IP address (layer 3) so they can be managed through IP or higher layer protocols.

[ATTRIBUTES]

S	?	n/a	n/a	n/a	n/a	9O67G	

NOTE: This recommendation does not call out specific methods or protocols for managing a bridge. The choice of management solution is left to vendors, but all bridges need to be IP addressable so that the specific management solution can be invoked over the network.

A.3.5 Device Recommendations: Interconnect
NC NID Ethernet Interconnect

[GUIDELINE] Network devices that interconnect Ethernet segments should provide switching of Ethernet frames and be compliant to 802.1D.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[1]	UVWWT	

NOTE: Ethernet switches forward Ethernet frames from one port to a single port based on the destination MAC address of the frame. This operating characteristic is highly desirable for QoS because the Ethernet frame will only be transmitted on the path necessary to reach its destination.

Devices that do not perform frame forwarding based on destination MAC address (some times called Ethernet Hubs) cause unnecessary collisions.

A.3.6 Device Recommendations: MoCA Bridge

A.3.6.1 NC NID MoCA Bridge: Connectivity

[GUIDELINE] MoCA Bridges should include Ethernet or 802.11 connectivity conformant to all [NC NID Ethernet:] or A.3.3.1/A.3.3.2 labeled requirements with bridging between the Ethernet, 802.11 and MoCA segments.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[97]	9XQ5T	

A.3.6.2 NC NID MoCA Bridge: MoCA Conformance

[GUIDELINE] MoCA Bridges should conform to MoCA Intermediate Device test plan requirements at the time the product is offered to the market.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[98]	S4RFJ	

NOTE: MoCA interoperability requirements will increase with time as new capabilities and features are specified by MoCA. When these capabilities are added to the MoCA certification test plans, MoCA implementations are expected to conform to them

A.3.6.3 NC NID MoCA Bridge: Upgradeability

[GUIDELINE] MoCA Bridges should be firmware updatable by the end user.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	n/a	XQ5T6	

A.3.6.4 NC NID MoCA Bridge: QoS Support

[GUIDELINE] MoCA Bridges should support DLNAQOS on all their network interfaces in accordance with the recommendations in A.3.6.5 and A.3.6.6.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	n/a	4RFJA	

A.3.6.5 NC NID MoCA Bridge: MoCA QoS Support

[GUIDELINE] If DLNAQOS is supported on a MoCA network interface by a MoCA Bridge, it should conform to all MoCA mandatory requirements.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[97]	Q5T6Z	

A.3.6.6 NC NID MoCA Bridge: MoCA Priority Mapping

A.3.6.6.1

[GUIDELINE] When bridging between the MoCA network interface and an 802.3 network interface, packets received on the MoCA network interface and transmitted on the 802.3 network interface should include the IEEE 802.1D user priority value in the IEEE 802.1Q header and the DSCP tag corresponding to the Priority of the received MoCA packets in accordance with the following table:

Table A.4 — MoCA Priority Mapping

MoCA Priority	IEEE 802.1D Priority	DSCP
(low)	0 BE	0x00
(medium)	5 VI	0x28
(high)	7 NC	0x38

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[1] [2] [32] [97]	RFJA4	

A.3.6.6.2

[GUIDELINE] When bridging between the MoCA network interface and an 802.3 network interface, packets received on the 802.3 network interface and transmitted on the MoCA network interface should include the MoCA Priority corresponding to the IEEE 802.1D user priority value in the IEEE 802.1Q header tag of the received 802.3 packets in accordance with the following table:

Table A.5 — MoCA Access and IEEE 802.1D Priority

IEEE 802.1D Priority	DSCP	MoCA Priority
1 BK	0x08	(low)
2 -	0x10	(low)
0 BE	0x00	(low)
3 EE	0x18	(low)
4 CL	0x20	(medium)
5 VI	0x28	(medium)
6 VO	0x30	(high)
7 NC	0x38	(high)

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[1] [2] [32] [97]	5T6ZG	

A.3.6.6.3

[GUIDELINE] When bridging between the MoCA network interface and an 802.3 network interface and an 802.3 packet is received that does not contain an 802.1Q tag, the MoCA bridge should look at the DSCP tag and map that to a MoCA Priority in accordance with the table in A.3.6.6.2 and preserve the DSCP tag across the 802.3 and MoCA segments.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[1] [2] [32] [97]	FJA4S	

A.3.6.6.4

[GUIDELINE] If a MoCA Bridge receives an 802.3 packet that does not contain an 802.1Q or a DSCP tag, the packet should be passed through to the MoCA interface unmodified without the addition of any priority tagging.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[1] [2] [32]	T6ZGL	

					[97]		

A.3.6.6.5

[GUIDELINE] If a MoCA Bridge receives a MoCA packet that is not tagged with a MoCA Priority, the packet should be passed through to the 802.3 interface unmodified without the addition of any priority tagging.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[1] [2] [32] [97]	JA4S9	

A.3.7 Device Recommendations: HPNA Bridge

A.3.7.1 NC NID HPNA Bridge: Connectivity

[GUIDELINE] HPNA Bridges should include Ethernet or 802.11 connectivity conformant to all [NC NID Ethernet:] or A.3.3.1/A.3.3.2 labeled requirements with bridging between the Ethernet, 802.11 and HPNA segments.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[59]	Q7INV	N

A.3.7.2 NC NID HPNA Bridge: HPNA Conformance

[GUIDELINE] HPNA Bridges should conform to HPNA Intermediate Device test plan requirements at the time the product is offered to the market.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[102]	RZ3LO	N

NOTE: HPNA interoperability requirements will increase with time as new capabilities and features are specified by HPNA. When these capabilities are added to the HPNA certification test plans, HPNA implementations are expected to conform to them.

A.3.7.3 NC NID HPNA Bridge: Upgradeability

[GUIDELINE] HPNA Bridges should be firmware updatable by the end user.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	n/a	DJTQH	N

A.3.7.4 NC NID HPNA Bridge: QoS Support

[GUIDELINE] HPNA Bridges should support DLNAQOS on all their network interfaces in accordance with the recommendations in A.3.7.5 and A.3.7.6.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	n/a	H45A3	N

A.3.7.5 NC NID HPNA Bridge: HPNA QoS Support

[GUIDELINE] If DLNAQOS is supported on a HPNA network interface by a HPNA Bridge, it should conform to all HPNA mandatory requirements.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[59]	I2BVZ	N

NOTE: QoS support is optional but if supported should conform to HPNA requirements. The HPNA specification provides the base level QoS specification for HPNA network devices.

A.3.7.6 NC NID HPNA Bridge: HPNA Priority Mapping

A.3.7.6.1

[GUIDELINE] When bridging between the HPNA network interface and an 802.3 network interface, packets received on the HPNA network interface and transmitted on the 802.3 network interface should include the IEEE 802.1D user priority value in the IEEE 802.1Q header and the DSCP tag corresponding to the Priority of the received HPNA packets in accordance with the following table:

Table A.6 — HPNA Priority Mapping

HPNA Priority	IEEE 802.1D Priority	DSCP
0	1 BK	0x08
1	2 -	0x10
2	0 BE	0x00
3	3 EE	0x18
4	4 CL	0x20
5	5 VI	0x28
6	7 NC	0x38
7	6 VO	0x30

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[1] [2] [32] [59]	NBBZC	N

A.3.7.6.2

[GUIDELINE] When bridging between the HPNA network interface and an 802.3 network interface, packets received on the 802.3 network interface and transmitted on the HPNA network interface should include the HPNA Priority corresponding to the IEEE 802.1D user priority value in the IEEE 802.1Q header tag of the received 802.3 packets in accordance with the following table:

Table A.7 — HPNA Access and IEEE 802.1D Priority

IEEE 802.1D Priority	DSCP	HPNA Priority
1 BK	0x08	0
2 -	0x10	1
0 BE	0x00	2
3 EE	0x18	3
4 CL	0x20	4
5 VI	0x28	5
6 VO	0x30	7

IEEE 802.1D Priority	DSCP	HPNA Priority
7 NC	0x38	6

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[1] [2] [32] [59]	PY994	N

A.3.7.6.3

[GUIDELINE] When bridging between the HPNA network interface and an 802.3 network interface and an 802.3 packet is received that does not contain an 802.1Q tag, the HPNA bridge should look at the DSCP tag and map that to a HPNA Priority in accordance with the table in A.3.7.6.2 and preserve the DSCP tag across the 802.3 and HPNA segments.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[1] [2] [32] [59]	J67A4	N

A.3.7.6.4

[GUIDELINE] If a HPNA Bridge receives an 802.3 packet that does not contain an 802.1Q or a DSCP tag, the packet should be passed through to the HPNA interface unmodified without the addition of any priority tagging.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[1] [2] [32] [59]	NJT3K	N

A.3.7.6.5

[GUIDELINE] If a HPNA Bridge receives a HPNA packet that is not tagged with a HPNA Priority, the packet should be passed through to the 802.3 interface unmodified without the addition of any priority tagging.

[ATTRIBUTES]

S	R	n/a	n/a	n/a	[1] [2] [32] [59]	WAACF	N

Annex B
(informative)

Basic Tuner Representation

B.1 General

A Tuner is a component of a server device that makes audio and video content available to a Rendering Endpoint. This content can come from an audio or AV tuner. A key characteristic of a Tuner is the ability to decode or demultiplex a single media stream from a number of available audio or AV streams. Note that in this annex we refer to the abstract entity represented on the home network as a Tuner (capitalized) and the physical building block inside the server doing the decoding and demultiplexing as a tuner (without capitalization).

B.2 Tuner Objects

The Tuner is represented as a CDS container (object.container) object. If a Content Source has two or more identical tuners (for example a device with two NTSC analog tuners), each tuner can be represented as a separate container object, or these tuners can be represented as a single container. However, a single Tuner can present content from multiple sources (e.g., an STB that provides Satellite and Terrestrial broadcast content), provided each channel can be uniquely selected. A Basic Tuner container should have an informative name that enables a consumer to easily distinguish the tuner. This could be based on the type of tuner. A Basic Tuner container shall have a <dlna:containerType> property with a value of "Tuner_1_0" to allow control points to differentiate them from other Container types.

B.3 Channel Objects

B.3.1 General

A Tuner makes its content discoverable as one or more channels that are represented as CDS videoBroadcast (object.item.videoItem.videoBroadcast) or audioBroadcast (object.item.audioItem.audioBroadcast) items. Each Basic Tuner container should contain a videoBroadcast or audioBroadcast item to represent each tunable (or selectable) channel. A Basic Tuner container should contain only videoBroadcast or audioBroadcast items, or both. It may also contain other objects that are directly related to the Tuner device or a specific channel. Control points should gracefully ignore any items that they do not understand.

B.3.2 Channel Order

These CDS Broadcast items should be presented in the order that best represents the order that channels are typically presented to users. This allows a control point to perform "up channel" and "down channel" operations by selecting the next or previous CDS Broadcast item, respectively. The control point should utilize the order of the Broadcast items within the Container's XML element to determine this order. Depending on the type of Tuner, this might be ascending broadcast frequency, logical channel number assigned by a cable operator or satellite providers, etc. In certain regions, channels are typically selected by the user from a set or list of user assigned channels, often called "presets". In these applications the Server Device can choose to present the CDS Broadcast items in the order the user has configured the presets (see guideline 7.4.1.4.15.6).

B.3.3 Channel Number

Wherever possible, the Server Device should present a Channel Number for each CDS Broadcast item using the channelNr (upnp:channelNr) property. This allows the user to directly select the desired channel by direct entry, rather than relying solely on "channel up" and "channel down" actions.

The UPnP namespace currently does not provide a subChannelNr property that makes representation of some channel numbers difficult because the fact that the channelNr property is restricted to integer values. Digital Television broadcasts commonly provide a multiple-program Transport Stream within a single radio-frequency channel, and these programs are commonly referred to as "subchannels". At this time, it is up to the implementer to decide how to best represent subchannel numbers as there is no subChannelNr property in the upnp namespace. In the case of broadcast sources where there exists a "primary" subchannel, an implementer could create a CDS Broadcast item representing the "primary" subchannel using the main channel number (to preserve the user expected channel number order), then a set of CDS Broadcast items representing the subchannels, can be exposed by the DMS using channel numbers that are vendor specific. For example, an over-the-air ATSC broadcast on radio-frequency Channel 40 with four subchannels, with licensed Channel Number 7, could have a primary CDS Broadcast Item with a channelNr value of 7 and four additional CDS Broadcast Items with channelNr values of 900, 901, 902, and 903 for each subchannel.

If the Channel Number represents a preset number, the range should reflect the numbering scheme normally presented to the user. This will typically be an ordinal number sequence (see guideline 7.4.1.4.19.1).

B.3.4 Channel Name

Wherever possible, the Server Device should present a Channel Name for each CDS Broadcast item using the channelName (upnp:channelName) property. Examples of recommended names are station identification (KOIN, FM 101.9, etc.) or network affiliation. The channelName property should not represent program content. In addition, the channelName property should be unique across all CDS Broadcast items in the Basic Tuner container. For example, if a tuner was able to present both a Standard Definition and a High Definition broadcast of the National Cartoon Network (NCN) channel, they should be named "NCN" and "NCN HD", respectively to preserve uniqueness. The Channel Name should reflect the subchannel number where appropriate. For example, a channelName of "Channel 40-1", "NCN-1", or "KGW-1", etc. would be appropriate for an over-the-air ATSC CDS Broadcast item (see guideline 7.4.1.4.20).

B.3.5 Channel Title

The Channel Title is represented in the dc:title property, which all CDS items shall have. In decreasing order of preference it should describe the program contents (i.e. "History of Cartoons"), the channelName information ("NCN"), or channelNr information ("Channel 6") (see guideline 7.4.1.4.18).

B.4 Accessing a Tuner Channel

A Rendering Endpoint accesses a tuner channel by establishing a connection to the URI of the resource associated with the CDS Broadcast item. If the Content Source accepts the connection, it tunes to the channel represented by the CDS Broadcast item, and the channel's content is streamed to the Rendering Endpoint. A Content Source may allow more than one Rendering Endpoint to connect to a single CDS Broadcast item (streaming identical content to all connections). If multiple connections to a tuner are allowed, it is up to the implementers to define arbitration logic to handle multiple Rendering Endpoints attempting to establish connections to different CDS Broadcast items (requesting two or more different channels simultaneously). A Content Source should refuse such connection requests that cannot be accommodated and return an error code of 503 (Service Unavailable) for either the HTTP transport or the RTP transport using RTSP. A separate transport connection shall be established between each Serving and Rendering Endpoint even though identical content will be sent over each connection.

A typical scenario for a device incorporating both a Rendering Endpoint and control point component that interacts with a Tuner occurs in the following manner. The control point component presents the available channels to the user as they are exposed by a CDS. When the user selects a specific channel for viewing, the Rendering Endpoint component issues an HTTP Get to the Content Source for the URI of the selected channel's content to initiate streaming. When the user wishes to change channels, the Rendering Endpoint component closes the existing HTTP connection, and then issues a new HTTP GET to the Content Source for the URI of the new channel's content.

Implementers should note that there is no feedback mechanism to notify a control point or Rendering Endpoint that the current tuner channel has been changed by another control point or a local user. Once a Rendering Endpoint has established an HTTP connection with the Content Source to stream the Channel content, and later the Content Source changes the "current" channel, the Content Source should stop streaming content and close the HTTP connection to indicate to the Rendering Endpoint that the channel is no longer the "current" channel. A Rendering Endpoint may terminate an HTTP connection at any time that it no longer wishes to receive the broadcast content.

Rendering Endpoints should be aware of the buffering requirements that live broadcast content places on the Content Source. Due to the possible network congestion, the server will need to buffer any temporary differences in the streaming rates between the incoming broadcast stream and the rate that the Rendering Endpoint accepts data over the network. If the server is unable to buffer any difference in rates, some of the data in the incoming broadcast stream will be lost. To avoid such data loss, Rendering Endpoints should be designed to accept data from network with an average rate equal to the live broadcast. Rendering Endpoints should also be designed to accept live broadcast content as a continuous stream, rather than a series of burst transfers. Note that this does not prevent a Rendering Endpoint from buffering content at the beginning of the streaming session, or changing the amount of content buffered at the Rendering Endpoint during the session, to account for the normal (and often dynamic) delays in HTTP network traffic.

B.5 Tuner Example

The following XML document fragment shows a Server Device with two tuners; an NTSC TV Tuner and an FM Radio Tuner. Note that the NTSC Basic Tuner container utilizes channel numbers based on broadcast channels while the FM Basic Tuner container illustrates ordinal channel numbers representing presets.

```
<DIDL-Lite
   xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/"
   xmlns:dlna="urn:schemas-dlna-org:metadata-1-0/">
   <!-- Root Container -->
      <!--(NOTE: XML Comments prohibited per 7.3.2.30 and are only included for
clarity) -->
   <container id="0" parentID="-1" restricted="1" childCount="2">
      <dc:title>DLNA Device</dc:title>
      <upnp:class>object.container</upnp:class>
      <!-- NTSC TV Basic Tuner container -->
      <container id="1" parentID="0" restricted="1" childCount="2">
         <dc:title>NTSC TV Tuner</dc:title>
         <upnp:class>object.container</upnp:class>
         <dlna:containerType>Tuner_1_0</dlna:containerType>
         <!-- NTSC TV Channels -->
         <item id="1-1" parentID="1" restricted="1">
            <!-- Full Description -->
            <dc:title>Cartoons, Cartoons, Cartoons</dc:title>
            <upnp:class>object.item.videoItem.videoBroadcast</upnp:class>
            <upnp:genre>Movie</upnp:genre>
            <upnp:channelNr>2</upnp:channelNr>
            <upnp:channelName>PBS</upnp:channelName>
            <res protocolInfo="http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC">
             http://192.168.0.20:58849/Tuner1/ch2.mpg
            </res>
         </item>
```

```
        <item id="1-2" parentID="1" restricted="1">
            <!-- Minimal Description -->
            <dc:title>Channel 4</dc:title>
            <upnp:class>object.item.videoItem.videoBroadcast</upnp:class>
            <upnp:channelNr>4</ upnp:channelNr>
            <res protocolInfo="http-get:*:video/mpeg:DLNA.ORG_PN=MPEG_PS_NTSC";
            DLNA.ORG_FLAGS=85100000000000000000000000000000>
             http://192.168.0.20:58849/Tuner1/ch4.mpg
            </res>
        </item>
    </container>
    <!-- FM Radio Basic Tuner container -->
    <container id="2" parentID="0" restricted="1" childCount="3">
        <dc:title>FM Radio Tuner</dc:title>
        <upnp:class>object.container</upnp:class>
        <dlna:containerType>Tuner_1_0</dlna:containerType>
        <!-- FM Radio Channels -->
        <item id="2-1" parentID="2" restricted="1">
            <!-- Preset #1 -->
            <dc:title>FM 89.9</dc:title>
            <upnp:class>object.item.audioItem.audioBroadcast</upnp:class>
            <upnp:channelNr>1</upnp:channelNr>
            <upnp:channelName>FM 89.9</upnp:channelName>
            <res protocolInfo="http-get:*:audio/L16:DLNA.ORG_PN=LPCM;
             DLNA.ORG_FLAGS=85100000000000000000000000000000">
             http://192.168.0.20:58849/Tuner2/ch1.L16
            </res>
        </item>
        <item id="2-2" parentID="2" restricted="1">
            <!-- Preset #2 -->
            <dc:title>FM 101.9</dc:title>
            <upnp:class>object.item.audioItem.audioBroadcast</upnp:class>
            <upnp:channelNr>2</upnp:channelNr>
            <res protocolInfo="http-get:*:audio/L16:DLNA.ORG_PN=LPCM">
             http://192.168.0.20:58849/Tuner2/ch2.L16
            </res>
        </item>
        <item id="2-3" parentID="2" restricted="1">
            <!-- Preset #3 -->
            <dc:title>FM 95.5</dc:title>
            <upnp:class>object.item.audioItem.audioBroadcast</upnp:class>
            <upnp:channelNr>3</upnp:channelNr>
            <res protocolInfo="http-get:*:audio/L16:DLNA.ORG_PN=LPCM">
             http://192.168.0.20:58849/Tuner2/ch3.L16
            </res>
        </item>
    </container>
  </container>
</DIDL-Lite>
```

Annex C
(informative)

UPnP Devices with Multiple Network Interfaces

C.1 Representation at the UPnP Device Level

This annex describes the subtleties and the intent behind the DLNA Home Networked Device Interoperability Guidelines for DLNA devices that simultaneously use multiple network interfaces. Readers should be familiar with the language of the following guidelines: 7.3.2.27 DDC UPnP Multi Homing Rules and 7.4.1.4.5 MM DIDL-Lite Content: Multiple Points of Accessibility. This annex summarizes two problems: how to represent a UPnP device on multiple network interfaces and how to represent content available on multiple network interfaces. Although they are separate issues, the way a vendor solves the second problem will depend largely on how the first problem is solved. In the paragraphs below, much of the text will describe scenarios with two network interfaces for example purposes. The number of supported interfaces for UPnP devices may be more than two.

Currently, there are two primary techniques for representing UPnP device on multiple network interfaces. The first technique is for the UPnP device to represent itself as multiple UPnP devices at the UPnP network layer, by using different UDN values for each discoverable UPnP device, with each UPnP device bound to a specific network interface. Figure C.1 describes this concept, with one logical UPnP device advertising two UPnP AV MediaServers (DMS devices). Each UPnP AV MediaServer also has a different UDN. Furthermore, through guideline 7.3.2.27.4, control points also obtain the correct IP address for each logical UPnP device.

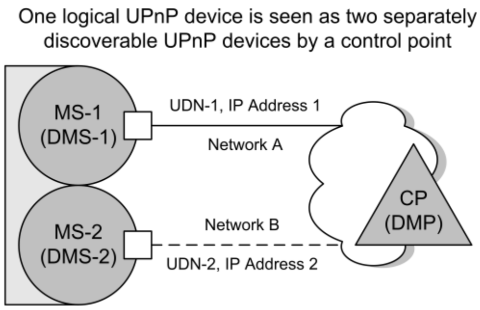


Figure C.1 — UPnP Device Representation

An important observation is that Network A and Network B can be bridged or completely separate networks. Another important clarification is that MS-1 and MS-2 are not part of he same UPnP device hierarchy. Equivalently, control points find MS-1 and MS-2 in separate device description files. For all intents and purposes, a control point that discovers MS-1 and MS-2 will not be able to conclude that both UPnP devices are part of the same product. Regardless of the topology, the only conclusions that a control point can make about the two UPnP devices is whether the (logical) UPnP devices are on the same UPnP network.

- If the control point sees UDN-1 and UDN-2 on the same network interface, then MS-1 and MS-2 are on the same UPnP network.
- If the control point sees UDN-1 and UDN-2 on different network interfaces, then MS-1 and MS-2 are on different UPnP networks.

Although a control point might not be able to identify the discoverable UPnP devices as part of a common logical UPnP device, additional meta-information might allow the user to make such a conclusion. For example, DMS-1 might have a UPnP friendly name of "Living Room Server (Wired)" and DMS-2 might have a friendly name of "Living Room Server (Wireless)". Of course, the friendly name for both UPnP devices could be identical, such as "Living Room Server". The Interoperability Guidelines do not make any recommendations or set requirements about the friendly names of UPnP devices because rules on meta-information depend more on philosophy and are less about protocol interoperability.

Lastly, even though the Interoperability Guidelines do not specifically state guidelines describing this type of behavior, the implementation technique is understood to be acceptable. The guidelines are worded to allow representation of a logical UPnP device through multiple, discoverable UPnP devices. The primary reason why this implementation technique is not described in guidelines is that it is virtually impossible for a UPnP control point to detect that two discoverable UPnP devices represent a logical UPnP device.

The other technique for representing UPnP devices on multiple network interfaces is to have the UPnP device report the same UDN on multiple network interfaces. Figure C.2 describes this concept.

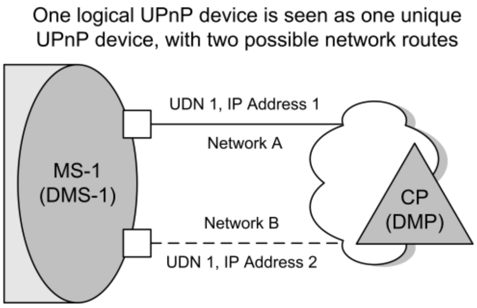


Figure C.2 — UPnP Device on Multiple Networks

Just like Figure C.1, Network A and Network B may be bridged or separate networks. In this type of implementation, a control point discovers only one UPnP device instead of multiple UPnP devices. However, the control point might see multiple IP addresses, depending on the network topology, allowing the control point to generally conclude one or more of the following:

- If the control point sees IP address 1 and IP address 2 on the same network interface, then the UPnP device is on one network with two different addresses.
- If the control point sees IP address 1 and IP address 2 on separate network interfaces (within 10 seconds of each other), then the UPnP device is on two different UPnP networks.
- If the control point sees IP address 1 and sees IP address 2 after 10 seconds, then the control point can conclude that the UPnP device has IP address 2 as the more reliable IP destination.
- If the control point sees IP address 1 and sees IP address 2 within 10 seconds, then the control point can conclude that the UPnP device has two IP destinations that seem equally reliable.

The advantage of using this technique is that the control point knows for sure that there is only one UPnP device. This allows the user interface of a control point to report one UPnP device instead of reporting multiple UPnP devices.

The Interoperability Guidelines focus mostly on what the UPnP devices can or shall do about multiple network interfaces. The Interoperability Guidelines do not specify any mandatory behavior for a control point because vendors believe that a variety of techniques can be used to present UPnP devices to a user. Guidelines 7.3.2.27.4 and 7.3.2.27.5 provide some ideas about what a control point can do, but vendors will need to design their control point taking into account many factors that are not discussed in the Interoperability Guidelines.

C.2 Representation at the CDS Level

Just as there are two primary techniques for representing UPnP devices with multiple network interfaces, there are also two primary techniques for representing content exposed by a UPnP AV MediaServer.

The first technique shown in Figure C.3 for representing content available on multiple network interfaces builds on the first technique for representing UPnP devices. Essentially, the DMS implementation uses multiple logical DMS representations and each DMS exposes URI values that a control point can treat as routable URI values from that DMS, as described below.

- One logical UPnP AV MediaServer represents itself as multiple discoverable UPnP AV MediaServers (DMS devices).
- Each discoverable UPnP AV MediaServer is associated with one network interface.
- Each discoverable UPnP AV MediaServer has a different UDN value.
- Each discoverable UPnP AV MediaServer exposes content that is "treated as or assumed to be routable" from the associated interface. Essentially, a control point can assume that there exists a network route from the control point to any of URI values' network addresses returned by the DMS.

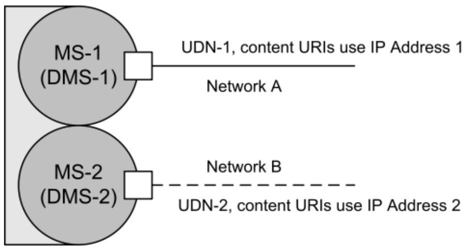


Figure C.3 — Representation at the CDS Level

When a control point finds content on this type of a DMS implementation, the control point can safely assume that a network route exists between the control point and each of the returned URI values. This assumption is an essential part of the "treated as or assumed to be routable" clause of guideline 7.4.1.4.5.1. A control point that finds content on DMS-1 will never see URI values that use a Network B address. Likewise, a control point that finds content on DMS-2 will never see URI values that use a Network A address. Although the content on both DMS-1 and DMS-2 might be the same content, control points cannot make this conclusion because DMS-1 and DMS-2 use different UDN values, forcing the control point to assume that they are two different DMS endpoints.

The second technique shown in Figure C.4 for representing content available on multiple network interfaces builds on the second technique for representing UPnP devices. Essentially, the DMS implementation uses one DMS representation, and the URI values that are reported depend on the Filter argument and on the network interface that received the SOAP request, as described below.

- The logical UPnP AV MediaServer represents itself with a single discoverable UPnP AV MediaServer (DMS device).
- The discoverable UPnP AV MediaServer is associated with all available network interfaces.

- The discoverable UPnP AV MediaServer reports the same UDN value on each network interface.
- If the discoverable UPnP AV MediaServer receives a CDS:Browse or CDS:Search request and the Filter argument does not have the ALLIP value, then it returns all URI values for the network interface that received the SOAP request. Essentially, a control point can assume that there exists a network route from the control point to any of the URI values' network addresses, which are returned by the DMS.
- If the discoverable UPnP AV MediaServer receives a CDS:Browse or CDS:Search request and the Filter argument has the ALLIP value, then the UPnP AV MediaServer responds with all URI values, regardless of whether the URI is associated with the interface that received the SOAP request.

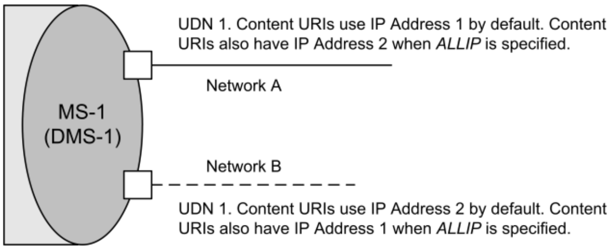


Figure C.4 — Content URIs over Multiple Networks

For this type of implementation, a control point that does not use the ALLIP value in the Filter argument can safely assume that a network route exists between the control point and each of the returned URI values. This assumption can be made because of guideline 7.4.1.4.5.1, which requires DMS-1 to never report URI values that have a Network B address, unless ALLIP is used. Likewise, DMS-2 can never report URI values that have a Network A address, unless ALLIP is used.

However, when the ALLIP value is used in the Filter argument, a control point will get all of the URI values, regardless of their network address values. Although not important for transactions between a DMS and DMP, this capability becomes important for future use cases.

C.3 Understanding the "treated as or assumed to be routable" Clause

To build on the examples in the previous subclause, MS-1 is on Network A with IP Address 1. When a control point finds content on MS-1, the control point will receive <res> URI values with IP Address 1. The control point will never see a <res> URI value with IP Address 2 when communicating with MS-1 because that would be a violation of guideline 7.4.1.4.5.1. One interesting aspect of the clause occurs when content is not served by the DMS implementation (i.e. it is advertised by the DMS but stored elsewhere). Technically an Internet-sourced URI is not prohibited so long as the URI is routable from the Internet to the local network. Since this condition is difficult to guarantee (e.g. an Internet service is down) and many see the value of Internet-sourced content for the future, the DLNA Home Networked Device Interoperability Guidelines use this clause instead of explicitly stating "all URI values shall have the same network address." In the case where ALLIP is used, control points need to be careful about non-routable addresses.

Lastly, this clause applies to any IPv4 URI regardless of whether the content complies with a DLNA media format profile. In the case of non-IPv4 URI values, a DMS should always publish non-IPv4 URI values (e.g. IEEE 1394, etc.) because a DMP can determine routability from the ProtocolInfo value.

C.4 Multiple <res> elements

On the issue of multiple network interfaces, guidelines 7.4.1.4.5.3 and 7.4.1.4.5.4 recommend that a DMS publishes multiple <res> elements (of each CDS object) instead of duplicate CDS objects. The DLNA Home Networked Device Interoperability Guidelines do not specifically mention the use of multiple CDS objects because this behavior is legal for UPnP AV. However, building a DMS to report multiple CDS objects might result in a user interface displaying multiple entries, with duplicate metadata. Since lower resolution television screens have limited space, the DLNA recommends that vendors avoid this type of implementation. The use of multiple <res> elements is a better approach because it allows control points to determine that the same content is accessible on different networks, in different formats, or via different transports. Furthermore, control points can build better user interfaces.

Annex D
(informative)

Printer Support

D.1 Introduction

Note: This annex is written mainly to assist printing controller (+PR1+ and +PR2+ Device Capabilies as defined in 5.11 2 Box Printing System Usage and 5.12 3 Box Printing System Usage) implementers

Printers are unique devices and their support requires a different approach and understanding from other DLNA media devices. One of the biggest differences is the overall reliability of a printer; printers run out of media and ink and require regular user intervention to keep them functioning. This characteristic of printers requires a different architectural approach than other DLNA media devices. The good news is that everyone at one time or another has used a printer so the fundamental expectation of how they're supposed to work is well understood by developers and users. There are many aspects of printer support that go unnoticed and in a DLNA context there are new considerations that need to be taken into account in developing a printing solution. This annex introduces developers to the technical considerations required to support printers and also discusses some of the usability aspects of printing that are important for a good user experience.

D.2 Printer/Printing Considerations

In order to understand the unique considerations for printers and printing it is easiest to start with a use case. For the purposes of this discussion the use case that will be considered is printing collections of photos from a printer control point (i.e. a +PR1+, +PR2+). Although other use cases should be considered when developing a printing solution, (e.g., electronic program guides, email, web pages, etc...), photo printing is more interesting because it requires not only control of the print data but also control over the size and type of paper used to print on.

There are a couple of significant differences between a DMPr and what we're all used to in normal everyday PC-attached printers (Table D.1).

Table D.1 — DMPr Printer verses PC Attached Printer

	PC Attached Printer	DMPr
Location	In the case of a typical desktop printer it's usually in close proximity to the user. Traditional network printers are also a good model but their operation does not entail the diversity of media needed for printing photographs.	Likely to be in a different room than the +PR1+ or +PR2+. A DMPr will be connected into a DLNA home network, but will likely be in the form factor of a typical desktop printer. The model is similar to a network printer but with a great deal more user intervention required to load photo paper when needed.
Print Content	PC-based application printing and Web-based content, as well as photos, email, etc. PC-attached printers have a dedicated driver written for the printer that allows for precise document printing, typically referred to as WYSIWYG (What you see is what you get.)	The content is displayed on a TV set, hand held device, or a networked display. All DMPrs support the PrintEnhanced:1 service, so a single print application (driver) can be used across all DMPrs. Exact document reproducibility requiring capabilities such as precise character positioning is generally not possible.

In most cases a DMPr will also be a fully capable PC printer, networked or directly connected to a PC. A DMPr has the additional UPnP PrintEnhanced:1 service that allows it to receive print requests from DLNA and other UPnP-based devices.

D.3 Paper Considerations

In the photo printing use case, correct attention to paper handling is one of the most important issues. Photo prints have a significantly higher cost to the user due to expensive media and ink. A typical scenario is that the printer will have normal everyday plain paper loaded in its media tray at the time the user clicks the print button. What makes this even more interesting is that many printers do not have a means to sense the media type or size until the print action begins. In this scenario, if the desired output is a 4"x6" print on photo paper, a user might have to intervene to ensure the proper media is loaded. UPnP PrintEnhanced:1 defines a means to accomplish this. What is important to consider is that this scenario shall be taken into account and that a photo print job is not just a "launch and forget about it" action. The control point will have to communicate with the printer and indicate to the user to load 4"x6" photo paper. Some, but not all, printers have a local UI sufficiently capable of indicating to the user the need to change the media and, in this case, the UI on the +PR1+ or +PR2+ should automatically clear out any message when the printer resumes printing. To minimize the potential for errors and frustration, it is recommended that the user be prompted to insert the desired media before the print job is initiated (Table D.2).

Table D.2 — Printing Controller (+PR1+, +PR2+) UI Components

Action	Recommendation
Printer Discovery and Setup	Many of the options presented to the user will likely depend on the capabilities of the discovered printer. For example, the paper sizes available will impact the layout options displayed.
Select Printer	If more then one printer is discovered, somewhere in the user dialog the desired printer needs to be selected. This is generally not a one-time setup issue, users might want to select different printers depending on their needs for a particular print task.
View and Select Content	Ideally, printing what is displayed on the client device should be an option for the user in all situations. For example, through a "Print Screen" button. The ability to select multiple photos before clicking the print button is an important user interface capability when printing images, as is allowing the user to specify the desired size of the photos or the number of photos per page.
Select Paper Size and Layout	Paper selection and control can be a very involved topic considering the regional differences in paper sizes. For example in Japan Hagaki Cards are very popular, in the US 4"x6" is typical, and in Europe A6 is a common photo size. It might be desirable to filter the values returned by the printer and displayed to the user to the set of sizes pertinent to the locale. The capability to print borderless photos is a very compelling feature but is dependent on the printer and the media selected. UPnP PrintEnhanced:1 provides the capability to learn all the media sizes and types supported by the printer; the smallest margins the printer can render to; and whether or not borderless prints are supported for each media. However, it is important to remember that media which are supported are not necessarily loaded.
Print Preview	Because of the cost associated with a bad print this option is always a good idea.

Action	Recommendation
Monitor and Display Printer Status	Designing a great printing experience needs to take into account all of the exceptions that can and will happen. Monitoring and displaying the status of the printer is a step that is commonly overlooked because in the ideal situation it isn't needed. However, developers shall consider that in many situations users might not know how to diagnose and correct common problems without assistance from the UI. UPnP PrintEnhanced:1 provides all of the status information needed to enable a robust user experience. Below is a list of the more common exceptions and recommendations on how to handle them. Please review the UPnP PrintEnhanced:1 speccification for a complete list of possible printer errors.

Table D.3 — Printer Status - Response

Issue/Error	Corrective Action
Wrong Paper loaded - size and/or type. Out of paper	Provide a UI message to load the correct paper; continue, or cancel.
Printer error - e.g. Paper jam, out of ink.	Error message indicating what the problem is. In many cases the job will be aborted by the printer.
Warning - Low on ink, low on paper, etc.	Appropriate warning message.
Content not found	Appropriate warning message
Print progress - Long Print Jobs	Printing large numbers of photos can take a long time so providing a progress indicator is desirable.

Layout options are usually at the discretion of the implementer. Users enjoy different album page type layouts and also common photo sizes (e.g. 4"x6"). The following diagram illustrates a simple set of layout options for 8 ½" x 11" or A4 paper that can be used for simple album pages and that also allow the user to cut the photos out in standard sizes. Centering and cropping the photos so that the aspect ratio is maintained for standard photo sizes is an important consideration (Figure D.1). The control of the print content is accomplished through XHTML Print and CSS Print Profiles.

The UPnP web site contains an excellent set of XHTML photo templates that are available to us as is or as a template to build on. Please refer to XHTML-Print Photo Templates for UPnP PrintEnhanced:1 [54] for more information.

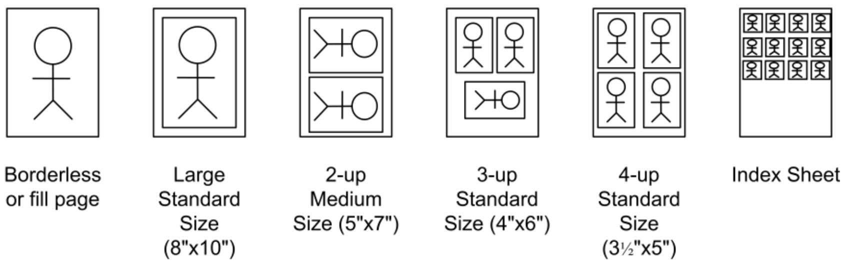


Figure D.1 — Photo Layout Options

D.4 Architecture

The architecture of a DMPr is based on the UPnP framework and can easily fit into the overall UPnP infrastructure used by other DLNA devices. The DLNA guidelines for discovery, actions and events, as well as data exchange using HTTP, apply to DMPr devices. The key technologies for DMPr devices are UPnP PrintEnhanced:1 and XHTML-Print plus CSS Print and CSS Print Enhanced Layout. UPnP PrintEnhanced:1 provides the actions and events for controlling the printer and submission of the print jobs. XHTML-Print coupled with the CSS Print Profiles comprise the Page Description Language used to control the content that goes on the page. The following diagram illustrates the above components on a DMPr (Figure D.2).

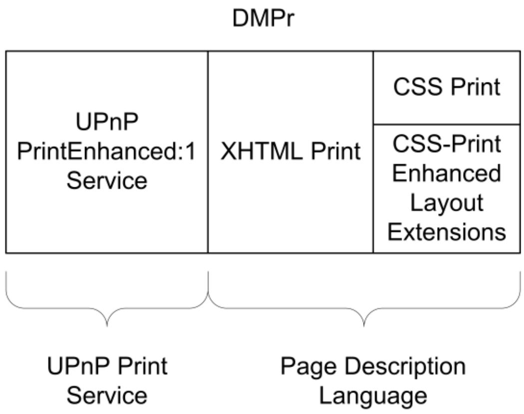


Figure D.2 — DMPr Architecture Components

Learning XHTML Print, CSS-Print, and CSS-Print Ehnhnaced Layout Extensions can be a very time consuming effort without having previous XHTML experience. One good place to start is the XHTML-Print/CSS Print Guidelines for PrintEnhanced:1 [77]. This document is very valuable to review to compose XHTML Print documents that will be more robust across different implementation of XHTML.

There are some very interesting features of PrintEnhanced:1 that bear further examination. The first feature is the ability to print content via a URI reference. A UPnP Control Point such as a +PR1+ can simply pass a URI (referencing content on the local network or the Internet) to the printer and the printer will fetch the XHTML document and process it. The +PR1+ or +PR2+ can ask the printer if it's connected to the Internet in advance of submitting the print job.

A second interesting feature of PrintEnhanced:1 is the ability to set up 'Critical Attributes' that shall be met by the printer in order for the print job to continue. An example of this is the media type.The +PR1+ or +PR2+ can tell the printer not to print unless it has the desired media loaded.

The third new feature of PrintEnhanced:1 is the ability to ask the printer the media sizes and types that it supports as well as the margin and borderless capability for each media type/size.

UPnP PrintEnhanced:1 contains 9 actions and approximately 34 variables that allow a +PR1+ or +PR2+ to control the printer. Two of the actions are deprecated so they are not mentioned here. The following (Table D.4) is a brief introduction to the actions in PrintEnhanced:1:

Table D.4 — UPnP PrintEnhanced:1 Actions Summary

Action	Summary
GetPrinterAttributesV2	Allows a +PR1+ or +PR2+ to determine various aspects of the Printer's current state plus an indication of whether or not the printer currently has an active connection to the Internet. The current JobId number and printer state are other examples.
CreateJobV2	This is another action that can be used to submit a print job. The actual XHTML-Print document is sent via an HTTP push from the +PR1+ or +PR2+. Although CreateJobV2 is mentioned here for completeness, DLNA guidelines indicate that CreateURIJob is the correct action for all DLNA printing.
CreateURIJob	Similar to CreateJobV2, this action is used to start a print job that pulls the XHTML-Print document from a provided URI. This is the action that DLNA guidelines require to be used for all DLNA print jobs.
GetJobAttributes	Allows a +PR1+ or +PR2+ to query the information related to the current job or any queued job.
GetMediaList	Provides a +PR1+ or +PR2+ the means to query the printer about the media sizes and types that it supports. Note that the sizes and types returned are not necessarily the ones currently loaded in the media tray.
GetMargins	Given the media size and type, this action returns the margins and a Boolean indicating whether or not the printer supports borderless printing for this media combination.
CancelJob	Allows a +PR1+ or +PR2+ to cancel any job, active or queued.

In addition to the actions there are 7 evented variables (Table D.5):

Table D.5 — Evented Variables

Event	Summary
PrinterState	Indicates a status of idle, processing, or stopped.
PrinterStateReasons	Useful for getting more information on why a printer might be stopped.
JobIdList	Useful for knowing how many jobs are in the queue and where a job is in relationship to the entire queue.
JobEndState	Indicates to a +PR1+ or +PR2+ how a print job ended and other details such as how many pages were printed.
JobMediaSheetsCompleted	This variable is evented every time the printer finishes a page. This is intended to be used to indicate to the user the progress of their job.
ContentCompleteList	Indicates to a +PR1+ or +PR2+ whether a printer has downloaded all of the content for the print job. This allows a +PR1+ or +PR2+ to know if the XHTML documents and referenced images can be deleted and/or if it is safe for the +PR1+ or +PR2+ to disconnect from the network. Intended for battery powered handheld devices that act as a +PR1+ or +PR2+. NOTE: There is a trade-off between an early exit from the network to save battery power and status information as the job is processed and completes. It is recommended that the user is provided an option to remain connected and monitor the job's progress
JobAbortState	Indicates to a +PR1+ or +PR2+ the reason why a print job was aborted.

D.5 Topology

The printable content for a DMPr can reside in many locations. XHTML-Print, in the same way as HTML, uses embedded URI's to reference images on the network. Printable images and other content can reside on multiple servers whose only requirement is that they can serve HTTP content. As mentioned previously, the URI's can even refer to content on the Internet.

One very subtle but important issue is that the content shall remain in place until the print job is complete or the JobId is listed in the ContentCompleteList state variable. Consider a multi-photo muti-page print job. If the photo on the last page is deleted or moved before the printer reaches that point in the print job, the job will not print to the user's satisfaction. In this scenario it is possible to indicate to the printer in the setup of the print job using 'critical attributes' (such as "image-layout" and "pdl-fidelity") to either abort or continue if this situation occurs.

Annex E
(informative)

Example Applications of the Uniform Client Data Availability Model

E.1 Uniform Client Data Availability Model Definitions

E.1.1 General

This annex clarifies the general applicability of the Uniform Client Data Availability Model (UCDAM). The annex describes the data accessibility assumptions for both Content Sources and Content Receivers. The UCDAM model strives for completeness by using examples derived from stored, converted, and live content streams. The model also accounts for caching of data by Content Receivers.

E.1.2 The Stream

In the most abstract sense, a stream is simply a data range of content, defined as [d_X, d_Y]. For content stored within a file, the data range for a content stream is fixed. This means that d_X and d_Y remain fixed over time. In some cases, the stream never ends, such that d_Y increases with time. For example, content described as "being sourced from a tuner" or "an infinite broadcast stream" are examples of infinite data streams.

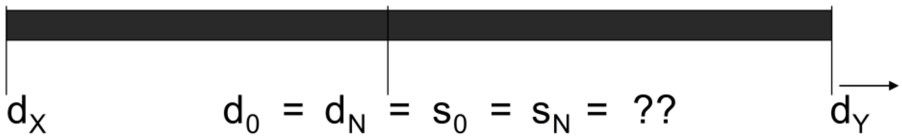


Figure E.1 — Abstract representation of a stream

Given a stream, there is one data range of primary interest: [s_0, s_N]. This data range represents what a Content Source can transmit. A secondary data range of interest is [r_0, r_N], which is a data range that supports random access operations on the Content Source. If enabled, this data range is always a equal [s_0, s_N].

Tangential to these data ranges is the [d_0, d_N] data range. This is the range of data that the Content Receiver has available to it from either local buffering or directly from the Content Source. This data range is not referenced in transport layer guidelines because Content Receivers have a wide range of options for local buffering techniques. Ultimately, the [d_0, d_N] data range is neither discoverable nor of any interest to Content Sources, so the focus of the guidelines remains on the Content Source data ranges.

Since the DLNA guidelines primarily focus on network transactions, the guidelines generally avoid distinctions between stored, converted, or live content. The guidelines focus more on a Content Receiver's ability to determine the Content Source's transport layer behaviors. Therefore it is important to remember that implementation details that are discussed in this annex are informational only. The examples serve to illustrate how vendors can implement against the guidelines based on the UCDAM. Although examples might cite a specific context (such as stored content vs live content) actual implementations may deviate from these examples while conforming to the normative guidelines.

E.1.3 Stored Content

In the simplest cases involving Streaming Transfer Mode operations on stored content (Audio-only or AV media class), Content Sources are generally able to access the entire stream and provide the entire data range to the Content Reciever. This means that the [s_0, s_N] data range is equal to the [d_X, d_Y] data range, and both are fixed data ranges.

Figure E.2 — A stored content stream

Random Access

There is a subtle aspect to consider regarding random access for all content, including stored content. The UCDAM makes no claims regarding a Content Source's ability to perform random access operations on the [s_0, s_N] data range. Consider a DMS that does not advertise content with the DLNA.ORG_OP parameter. This type of a DMS claims that the Range HTTP header is not supported at the transport layer. Although the DMS is able to transmit all of the data in [d_X, d_Y], the ability for the Content Receiver to randomly access the data is not available. This limitation can affect the ability for a DMP or DMR to support media operations like pause, pause-release, seek, and forwards and backwards scanning with the DMS.

Figure E.3 — Stream with no random access support

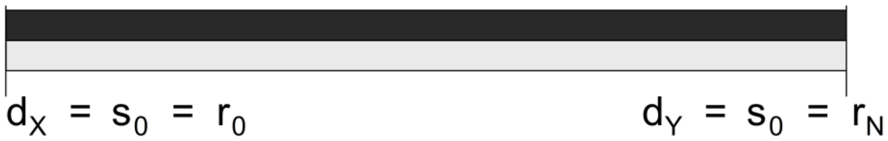


Figure E.4 — Stream with random access support

Random access operations are not limited to HTTP requests that involve the Range header. Other examples include HTTP requests with the TimeSeekRange.dlna.org header. The guidelines for RTP seek operations might also have dependencies on being able to do some form of random access.

E.1.4 Converted Content

Some DMS implementations are able to offer multiple versions of the same content. For purposes of discussion, assume that the original version of the content is a native version stored on the DMS. The additional formats made available by the DMS are called converted versions. Converted content is a convenient way for a DMS to provide support for baseline media format profiles when the original version is in an optional media format profile. Common forms of conversions include transcoding, transcaling, and transrating.

A Content Source that claims support for [r_0,r_N] data range is required to support random access on the entire range of [s_0, s_N]. The only time where this is computationally difficult is when performing random access operations when responding to requests with the Range header. This difficulty manifests itself primarily in the server having a (long) delay when responding to such requests. To assist Content Receivers in making intelligent decisions about using such requests, the guidelines allow the server to report a byte range that is a subset of $[r_0,r_N]$.

E.1.5 Live Content

Live content is more complex than stored or converted content because it opens up the possibility of infinite streams (increasing d_Y value) and introduces the concept that portions of a stream might never be available after a certain point in time (increasing s_0 and s_N values). This subclause describes a few variations that Content Sources could use when distributing a live stream. Please note that figures in this subclause are not drawn to scale.

During the initialization phase, the buffer exhibits the behavior of a growing buffer. In this phase, the Content Source has fixed values for d_X, s_0, r_0, and r_N, while having an increasing value for $d_Y = s_N$.

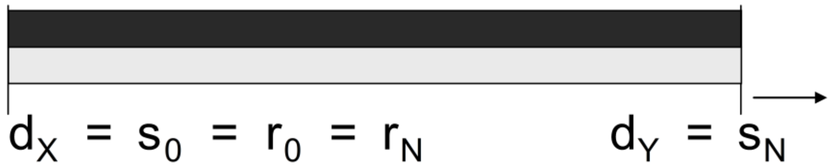


Figure E.5 — Live stream with growing buffer and no random access

If the Content Source is able to handle random access requests, then the model is slightly different. Given the UCDAM, it is possible for a Content Source to change the $[r_0, r_N]$ range after the transmission starts. In the figure below, the Content Source supports random access requests on the same range as the $[s_0, s_N]$ data range.

Figure E.6 — Live stream with growing buffer and random access

Since this is an infinite stream, the Content Source will eventually run out of local memory or storage for buffering. When this happens, the Content Source may exhibit a sliding window behavior.

Figure E.7 — Live stream with sliding buffer and random access support

The last variant is really an implementation detail that is hidden. Some live streams might be time delayed. Some implementations can use time delaying as a way of having sliding buffer from the very beginning of stream. In practice, there is no way to distinguish between a time-delayed live stream and other live streams. Therefore, it is functionally identical to the previous case. The only difference is the relationship between the normative s_N data position and the theoretical d_Y data position.

Figure E.8 — Time-delayed live stream with sliding buffer and random access support

E.2 UCDAM and Media Operations

E.2.1 General

This annex is an examination of the media operations and how Content Sources and Content Receivers need to account for data ranges.

E.2.2 Data Ranges

The DLNA guidelines do not mandate a particular buffering model on either Content Sources or Content Receivers. As a corollary, the guidelines do not prohibit Content Receiver implementations that cache content data. In practice, this means Content Sources have complete control over the behavior of their accessible content data range (i.e. $[s_0, s_N]$) and Content Receivers have complete control over the behavior of their accessible content data range (i.e. $[d_0, d_N]$).

For numerous Content Receivers, the $[s_0, s_N]$ and $[d_0, d_N]$ are effectively identical. This means that the Content Receiver only accesses data that the Content Source can provide at any given time. However, some Content Receivers are able to expand their $[d_0, d_N]$ data range by caching data. For example, a Content Receiver might be able to display data to the user that is no longer available from the Content Source.

Although the UCDAM takes data caching into account, DLNA guidelines do not specify how a Content Receiver determines its $[d_0, d_N]$ data range because it is considered an implementation detail that is out-of-scope. (This is the same reason why DLNA guidelines do not specify how a Content Source determines its $[s_0, s_N]$ buffer, relative to the theoretical $[d_X, d_Y]$ data range.

However, the role of guidelines is specifying interoperable behavior. To meet this objective, the DLNA guidelines defines media operations in terms of the $[d_0, d_N]$ data range because that is the data range that affects what a user will perceive. Simultaneously, the guidelines specify syntax and transport protocol requirements that govern transactions between Content Sources and Content Receivers.

E.2.3 Play Data Flow

The most basic streaming operation is the Play media operation. When a Contene Receiver initiates a Play media operation with a Content Source, the Content Source has to choose a starting playback position (dPlayStart), which is in the $[s_0, s_N]$ data range. In the stored content scenario, this maps to the first byte of the actual content file. The converted content scenario is very similar, except the converted content may be dynamically generated in response to a request.

In the case of live content, the Content Source generally uses the dLive position (at the time of receiving the Play request) as the dPlayStart value. Generally, the dLive position moves forward with time by being attached to an original broadcast source's live position. However the DLNA guidelines do not define a mandatory rate for updating the dLive position, so two Play requests could have the same dPlayStart even though they were made at different times. This leads to a corollary that dLive might be a time-shifted stream, although it is impossible for Content Receivers to know the size of the time-shift. The DLNA guidelines do not define a normative time-range for time-shifting, so it's theoretically possible for a Content Source (with a lot of local storage) to always choose the first byte of an always-increasing [s_0, s_N] data range as the dPlayStart.

In summation, vendors can assume two things. Content Sources have a lot of flexibility when choosing a dPlayStart position when responding to a Play request. Content receivers can rely on a convention that the first received byte maps to the "beginning of content" but the convention is not generally applicable to live content. Without precise, formal definitions for stored, converted, or live content, the guidelines can only rely on conventions, which might be sufficient since the typical convention for stored content is already in widespread use by both computing and consumer electronics devices.

E.2.4 Stop Data Flow

By convention, a device that can play can also stop. The Stop operation means that a user sees a stop in the playback. Devices implement a Stop operation by stopping the data flow at the network, with one subtle exception. Since the guidelines do not specify how a Content Receiver maintains its [d_0, d_N] data range, Content Receivers are permitted to continue data streaming from the Content Source. This operational policy is permissible because Content Receivers are the endpoints that initiate an end to data transmission.

E.2.5 Pause & Pause-release Data Flow

The DLNA guidelines define the pause and pause-release operations. Just like the case for the Stop operation, the guidelines have the general expectation that data flow will cease at the network, with exceptions made for Content Receivers that cache data.

In the cases for stored content, the Pause and Pause-release operations operate consistently by convention. Generally, a user initiates the Pause operation at some particular playback position (dPause). When the user initiates the Pause-release operation, playback resumes. The Content Source and Content Receiver determine the location dResume in the content stream where the transfer resumes. Depending on user preferences and the type of the stream, the Content Receiver might wish to continue the stream from where the Pause operation was initiated (i.e. dResume = dPause). In many cases however, the ability to resume can sometimes be affected by a Content Source's ability to randomly access data at the dResume position. For example, if the Content Receiver disconnected the TCP connection as part of a Pause operation, and it wishes to resume with dResume = dPause it should reconnect the TCP connection and perform a Seek operation to move to dPause. In order to satisfy this request, the Content Source would need to have dResume in the [r_0, r_N] data range in order to support this request. However, in the case of live or transcoded content, the Content Source might not be able satisfy this request. In this case, the Content Source and Content Receiver, given the knowledge of the Content Source's [s_0, s_N] data range, determine an appropriate location in the stream to continue the transfer. For example, for live content where $s_0=s_N$ the point at which the stream is continued is the current sample of the live stream.

Regardless of whether the content is stored, converted, or live, Content Receivers are always able to determine the supported transport layer features for the content. Therefore, the most consistent behavior is to disable the Pause operation when the Content Receiver detects that it cannot perform a Pause-release operation from the paused position.

E.2.6 Scan Operations

The DLNA guidelines define four types of scan operations: Fast Forward, Fast Backward, Slow Forward, and Slow Backward.

Since the UCDAM does not mandate a particular buffering model on the content source, there are no requirements about how the Content Source's [s_0, s_N] data range changes with time. In the case for stored and converted content, the [s_0, s_N] data range generally never changes. It remains fixed and represents the entire content binary. In the case of live content, the data range can grow, exhibiting a growing buffer. At other times the data range can appear to slide forward with time because the content source has a buffering limit. In some cases, the data range can even temporarily shrink because the content source buffers using a variable bitrate.

Despite the possible diversity of Content Source buffering models, Content Receivers need to handle one general problem that can happen with any of the scan operations: what does the Content Receiver do when a scan operation has reached the [d_0, d_N] boundary? (Usually, this problem is caused by a limited [s_0, s_N] data range.).

Annex F
(informative)

Auto-IP Developer Guidance

F.1

The purpose of this informative annex is to provide developer guidance on extending Auto-IP support for IP Stacks that have problems with full conformance to Auto-IP.

F.2 Introduction

The DLNA guidelines support two IP address allocation systems, DHCP and Auto-IP. DHCP supports the allocation of routable IP addresses for network environments that have multiple subnets, while Auto-IP allocates non-routable, link-local addresses. In various DLNA interoperability testing, it has been observed that some IP stack implementations have difficulty communicating between a device that has used DHCP to obtain its IP address and one that has used Auto-IP. This situation can arise during times when the DHCP server is offline. DHCP address leases expire at randomly spaced times, so some devices might need to renew their address leases while the DHCP server is offline. Not finding a DHCP server, the device will assign its own IP address with the Auto-IP protocol. Other devices, whose leases have not expired, will continue to use their DHCP assigned address. This mix of address allocation systems will persist until all of the leases have expired and every device has allocated Auto-IP addresses, or until the DHCP server is online again, and devices sense its presence and revert to using the DHCP server (as per the mechanism defined in the guidelines). This annex will provide additional guidance for device implementations to overcome the communication problems during this transition.

IP Mixed Network (Auto-IP and DHCP) depicts a simple network configuration where device A is using an IP address assigned by a DHCP server and device B has an Auto-IP allocated address. Auto-IP assigned addresses are in the range 169.254.0.0/16 while DHCP assigned addresses use other addressing ranges. Device A has been configured with IP address 192.168.1.100 and a subnet mask of 255.255.255.0. When it attempts to send a packet to device B with Auto-IP assigned address 169.254.21.113 it does not recognize that address as being on the local subnet and sends all IP packets to device B to the default gateway. Additionally, device B is not configured with a default gateway because it is using a link local address. When device B attempts to send a packet to device A, it cannot determine if the packet is bound for the local subnet or elsewhere. Because it does not have a gateway, the observed behavior for some implementations is for device B to hold all IP packets with a destination IP address different from the Auto-IP subnet. In this situation, device A and device B cannot communicate.

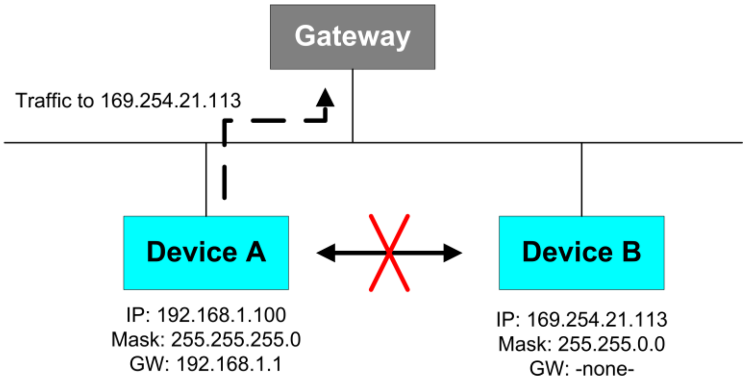


Figure F.1 — IP Mixed Network (Auto-IP & DHCP)

Some IP stack implementations have been observed to exhibit this behavior. While the Auto-IP specification version used by UPnP [47] requires that no packet with a link-local address as either the source or destination be sent to a gateway, it does not strictly require that a device attempt to send such a packet on the local subnet by default. It is also recognized that device manufacturers might not have the ability to change the IP stack implementations on their devices because they obtain the stacks from other third party software vendors. This annex suggests a simple mechanism to allow communication between devices with two different IP address types while not requiring software changes on either device.

F.3 Suggested solution

F.3.1 General

To overcome the problem of communication between devices that allocate their IP address with different systems, it is suggested to add new routes to each device that require packets bound to the other device to be sent on the local link. The advantage of such a solution is that it limits the effects of the IP address modification to the device changing its IP address. Note that these additional routes are correct for the types of traffic being sent on the local address, they make explicit the requirements of the Auto-IP specification to the routing software. Consequently, they can be used independent of the OS and system platform employed.

Disclaimer: although this mechanism has been verified on two major operating systems, this is not an absolute guarantee of success.

F.3.2 Route for an Auto-IP device sending packets

Devices with an Auto-IP address (e.g.: device B) should specify a default IP route for all traffic not part of the Auto-IP subnet (169.254.0.0/16). In contrast to a regular default route the new route does not direct default IP traffic to the gateway; rather it instructs the IP stack to forward all packets to an unknown subnet on the local link physical interface.

The consequence of such route is that traffic to any IP address (Auto-IP allocated, DHCP allocated or other IP addresses) is considered reachable on the local link. While this does not allow communication between devices that assign addresses with Auto-IP and devices on other subnets, this is not a limitation of the solution but rather, a limitation of Auto-IP itself. The Auto-IP specification requires that no packet to or from a device using an Auto-IP allocated address is to be forwarded to a gateway. This route simply makes that requirement explicit in the routing table.

Table F.1 — Auto-IP Route

Network destination	Netmask Interface	Gateway	
169.254.0.0	255.255.0.0	[OS dependent]	lan0
0.0.0.0	0.0.0.0	[OS dependent]	lan0

The table above shows the new route in bold.

F.3.3 Route for a DHCP device sending packets

Devices with a DHCP IP address (e.g.: device A) should specify a new IP route for all traffic bound to addresses within the Auto-IP address range. The expected behavior is to force the device with DHCP IP address to recognize a new subnet, namely the Auto-IP subnet. This route will be to specify that all packets bound for the Auto-IP subnet will be sent on the local address interface and not forwarded to the gateway. This is correct behavior since by definition all Auto-IP allocated address are on the local link.

Table F.2 — DHCP Route

Network destination	Netmask Interface	Gateway	
192.168.1.0	255.255.255.0	[OS dependent]	lan0
169.254.0.0	255.255.0.0	[OS dependent]	lan0
0.0.0.0	0.0.0.0	192.168.1.1	lan0

The table above shows the new route in bold.

F.4 Validation example using UPnP AV Applications

F.4.1 General

Using Microsoft Windows® 2000 Professional, Microsoft Window® XP Professional and Linux machines the following test has been conducted. Device A has a DHCP IP address with a fairly long lease time (in order to maintain the assigned IP address during the entire test). Once device A has received its DHCP IP address, the DHCP server is removed from the network and the second device (B) is connected and started. Without a discoverable DHCP server, device B assigns itself an Auto-IP address. The suggested routes are manually added using the appropriate command line application 'route'. UPnP AV applications are used to validate the communication between devices A & B. Device A runs an UPnP AV Media Server and device B runs an UPnP Media Renderer.

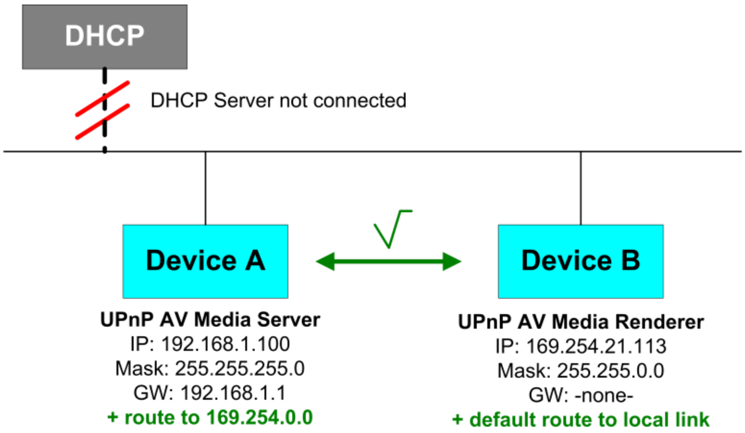


Figure F.2 — Communication in Mixed IP network.

The UPnP AV Media Server on device A and UPnP AV Media Renderer on device B broadcast their presence with their own IP addresses. With the help of the new routes both applications communicate correctly and transparently without any problems.

F.4.2 How to add a route on Windows 2000 & Windows XP?

The command line application 'route' allows you to manipulate network IP routing table. The command 'route add' is used to add a new route while 'route delete' is used to remove an existing route. For example, the first command listed below can be used on the Auto-IP device (device B in Figure F.2) to add a default route for all traffic to be placed on the local link. The second command listed below can be used on the DHCP device (device A in Figure F.2) to ensure that all traffic bound for Auto-IP device also is placed on the local link.

```
C:\> route add 0.0.0.0 mask 0.0.0.0 169.254.21.113            // device B
C:\> route add 169.254.0.0 mask 255.255.0.0 192.168.1.100     // device A
```

Please note that under Windows the IP address of the device itself (169.254.21.113 or 192.168.1.100) is used to specify a local link interface. The device's IP address is also used for the gateway parameter.

And the following commands respectively remove the same routes:

```
C:\> route delete 0.0.0.0 mask 0.0.0.0 169.254.21.113         // device B
C:\> route delete 169.254.0.0 mask 255.255.0.0 192.168.1.100  // device A
```

Alternatively the API functions CreateIpForwardEntry and DeleteIpForwardEntry from the Platform SDK*: IP Helper can also be used within an application to add and remove routing table entries.

Following tables show Windows routing table examples respectively for a device using an IP address assigned by a DHCP server (device A) and for a device using an Auto-IP address (device B).

Table F.3 — Windows routing table example for device w/DHCP Address

Active Routes:Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.1.1	192.168.1.100	30
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
169.254.0.0	255.255.0.0	192.168.1.100	192.168.1.100	30
192.168.1.0	255.255.255.0	192.168.1.100	192.168.1.100	30
Default Gateway:192.168.1.1				

Table F.4 — Windows routing table example for device w/Auto-IP Address.

Active Routes:Network Destination	Netmask	Gateway	Interface	Metric
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
169.254.0.0	255.255.0.0	169.254.21.113	169.254.21.113	30
169.254.21.113	255.255.255.255	127.0.0.1	127.0.0.1	30
Default Gateway:169.254.21.113				

F.4.3 How to add a route on Linux?

The command line application 'route' allows you to manipulate network IP routing tables. The command 'route add -net' is used to add a new network route while 'route del -net' is used to remove an existing network route. For example, the following commands respectively add the default route for the Auto-IP device and the route to Auto-IP subnet for the device with DHCP IP address:

```
user@host-B:# route add -net 0.0.0.0 netmask 0.0.0.0 eth0
user@host-A:# route add -net 169.254.0.0 netmask 255.255.0.0 eth0
```

And the following commands respectively remove the same routes:

```
user@host-B:# route del -net 0.0.0.0 netmask 0.0.0.0 eth0
user@host-A:# route del -net 169.254.0.0 netmask 255.255.0.0 eth0
```

Alternatively the system ioctl function in combination with socket IO Controls SIOCADDRT & SIOCDELRT can also be used within an application to add and remove routing table entries.

Following tables show Linux routing table examples respectively for a device using an IP address assigned by a DHCP server (device A) and for a device using an Auto-IP address (device B).

Kernel IP routing table

Table F.5 — Linux routing table example for device w/DHCP Address

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.1.0	*	255.255.255.0	U	0	0	0	eth0
169.254.0.0	*	255.255.0.0	U	0	0	0	eth0
default	192.168.1.1	0.0.0.0	UG	0	0	0	eth0

Kernel IP routing table

Table F.6 — Linux routing table example for device w/Auto-IP Address

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
169.254.0.0	*	255.255.0.0	U	0	0	0	eth0
default	*	0.0.0.0	UG	0	0	0	eth0

F.5 Installing Routes During Address Transitions

Please note that the network routing table shall be dynamically modified each time a new type of IP address is assigned to a device. When a device allocates a DHCP address, the default route for Auto-IP devices should be installed in the routing table. This route may remain in place across IP address transitions. When a device allocates an Auto-IP address, the default route for all traffic should be installed in the routing table, specifying the local link. This route shall be removed whenever the device transitions back to a DHCP address, and should be replaced by the gateway specification obtained via DHCP

Figure F.3 shows an example of the additional route (grey boxes) in the IP Address assignment flow.

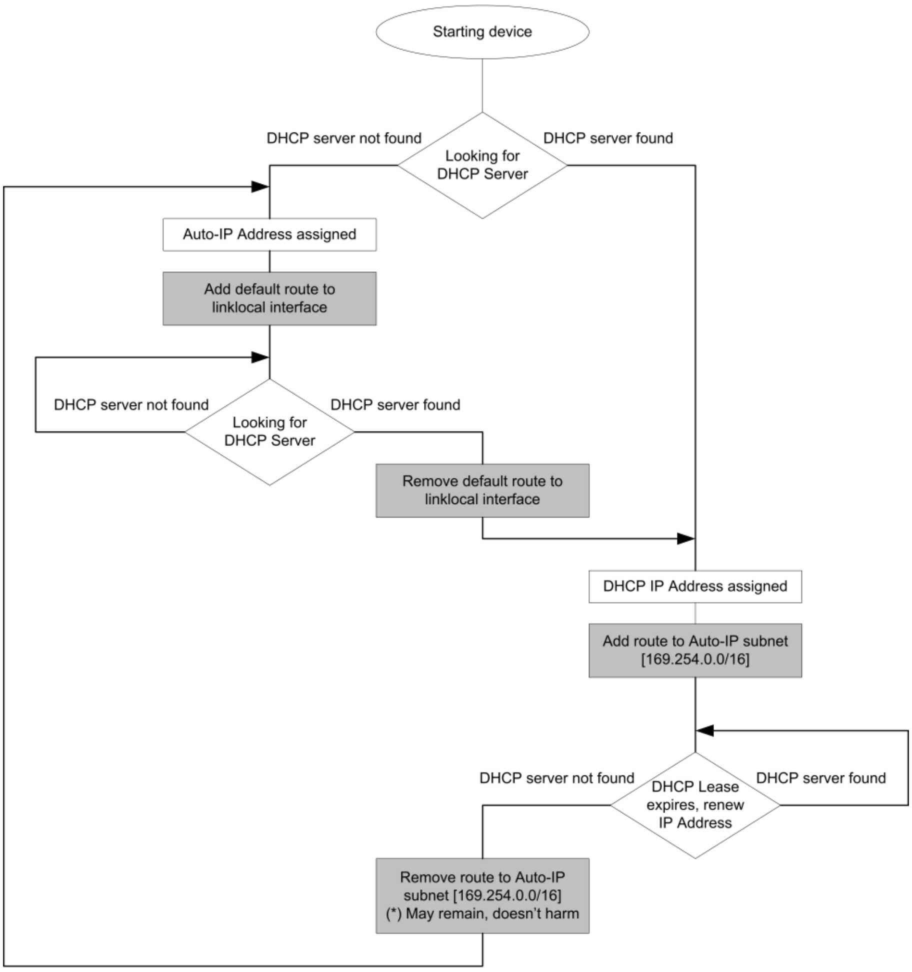


Figure F.3 — New routes in address transition flow

Annex G
(informative)

Mobile Network Connectivity and Power Saving Operation Principles

G.1 Mobile Device Interoperation with the Home Network

Mobile devices may use any method defined by the DLNA guidelines for connecting to the home network. If WLAN or Ethernet is used, the mobile device connects directly to the home network as specified for those connectivity methods. If the mobile device connects via Bluetooth, an additional device called the Mobile Network Connectivity Function (M-NCF) is needed to bridge between Bluetooth and rest of the home network. In practice, this bridging performed by the M-NCF is conformant to 802.1D [81]. In addition to bridging, the M-NCF provides additional functionality for power saving that is important for mobile, battery-powered devices and provides link level access control to the Bluetooth network.

Above the link layer, mobile devices are fully compliant with all DLNA guidelines for the home devices. Therefore, there are no separate guidelines for implementing IP, TCP/UDP, UPnP or higher layers on the mobile device. Readers should, however, notice that this considers only networking and UPnP as such. Media formats, for instance, are specified separately as part of the Media format document [56] and there are specific formats for mobile devices.

G.2 Bluetooth Security (NC-Security)

G.2.1 General

The M-NCF allows mobile devices like mobile phones, PDAs, and music players to easily attach to the home network. It also allows visitors like friends and relatives to connect to and use home network services. However, in order to keep a basic level of security and privacy, a set of guidelines has been defined that allow the M-NCF to provide adequate security based on link level mechanisms. These security requirements do not affect the functionality as seen on the network or higher layers, after link layer authentication and authorization has been completed. In these guidelines we define interoperability guidelines for providing easy access to the network for a home owner's devices, providing network access for a guest's devices, and also for revoking already granted access rights. These guidelines apply for Bluetooth connectivity only.

Please note that the term support has a definition specific to clause G.2. It means that the feature exists, but the feature is not required to be used.

G.2.2 Bluetooth Security Concepts

Authentication for Bluetooth is defined in [79] as the process of verifying 'who' is at the other end of the link. Authentication is performed for devices based on their Bluetooth Address (BD_ADDR). In Bluetooth this is achieved by the authentication procedure based on the mutually stored link key or by pairing (See below).

Authorization for Bluetooth is defined in [79] as the process of deciding if a device is allowed to have access to a particular service. This is where the concept of 'trusted' exists. Trusted devices (authenticated and indicated as "trusted"), are allowed access to services. Untrusted or unknown devices might require authorization based on user interaction before access to services is granted. This does not exclude the case where the authorization might be given by an application automatically. Authorization always includes authentication.

Pairing is the process used by Bluetooth to support authentication. If the device is unknown to an M-NCF, and there is no entry in the M-NCF's device database, the pairing process allows an M-NCF to establish a relationship with the corresponding mobile device. As a result of pairing, a pair-wise link key is created and stored in the device database of the M-NCF to be used to authenticate this mobile device the next time it requests access.

The Device Database is a concept widely used in Bluetooth security documents. The device database stores security related information about the mobile devices that have a relationship with the M-NCF. The relationship between an M-NCF and a mobile device is formed via the Bluetooth pairing process. This concept refers to a generic information storage space, and the values that shall be stored. It does not refer to mechanisms to create such a storage space. Implementation of the storage space is left open to the device manufacturer. For each mobile device, the device database is expected to hold the Bluetooth device address, the device name, a link key, and an indication of whether the device is trusted or not.

The Service database is also a Bluetooth concept. It is used to hold Bluetooth service related security information e.g. whether encryption, authorization or authentication is required for home network access. This is also concept without relation to any potential means of implementation.

G.2.3 Controlling access to home network via an M-NCF

Access control to the home network is based on Bluetooth security methods that are explained in relevant standards [82], [81], [81], and also in the Bluetooth security white paper [79]. In DLNA networking, access control for mobile devices is centralized in an M-NCF connected to the home network. Access control is based on classifying all devices into three classes; trusted, untrusted and unknown. Each of these classes shall have different treatment when requesting access. In the beginning, all devices are unknown and they have to be paired, and authorized for home access. Although the user interface can vary between different vendors certain basic operations shall be performed.

Since, the recommendation is that devices operate in security mode 2, the security procedure will not be initiated before a Bluetooth L2CAP connection is established[81]. At this point, device and service databases are queried to determine whether the device has an entry and its contents (identity, link key, trust level..). These steps enable a trusted device with valid link key to access the home network without any additional activity by the home owner. If the device is untrusted, the system should obtain the home owner's consent before access to home network is allowed. This can happen through some sort of user interface or even with just a single button. If the device requesting access is unknown to the M-NCF (i.e. there is no entry in the M-NCF's device database for this device), then Bluetooth pairing is required at this time, and the home owner should categorize the device as either trusted or untrusted.

As the home owner is able to control access to his/her home network and grant visitor access as needed, naturally, there will be the need to revoke this access. In these guidelines we have defined two mechanisms that allow vendors to build required features into their products. The mechanism of validity time allows the M-NCF to grant access on a temporary basis. After the validity time has passed, the information on the mobile device is automatically removed from the M-NCF's device database. When the mobile device next connects to the M-NCF, it will need to be granted access again.

Alternatively, revoking network access rights can be explicit. In this case, revocation is carried out by directly deleting the mobile device's records from the M-NCF's device database. Once the mobile device's access rights are revoked, all ongoing connections with this mobile device shall be terminated following the user action. This, in practice, requires that the home owner specifically select a certain device and cancel its access rights. The actual implementation is left for the vendors, but the mechanism to provide this feature shall be available in an M-NCF.

G.3 Network Connectivity Power Saving (NC-PS) Modes

Please note that the term support has a definition specific to clause G.2. It means that the feature exists, but the feature is not required to be used.

The NC-PS modes are defined as viewed from the IP-layer and above. These modes are abstractions of the underlying bearer-specific power saving mechanisms and expose an abstract NC-PS layer to application-level entities (e.g. future UPnP low-power proxies, applications, middleware), in order to hide from them link-layer complexities and bearer specific implementations. From an application point of view, it is enough to know these high level modes and how they affect the perceived network service. Each of the abstract NC-PS modes is instantiated through a mapping to the underlying bearer's power saving mechanisms. The NC-PS guidelines define how this mapping is done and the bearer parameters that allow it to meet the NC-PS mode connectivity requirements. It is noted that this version of the guidelines defines this mapping only when the underlying bearer is Bluetooth; future versions might address mappings to other wireless bearers supported by a mobile device. Figure G.1 illustrates the abstraction introduced by the NC-PS between the bearer-specific power saving mechanisms and application-level entities.

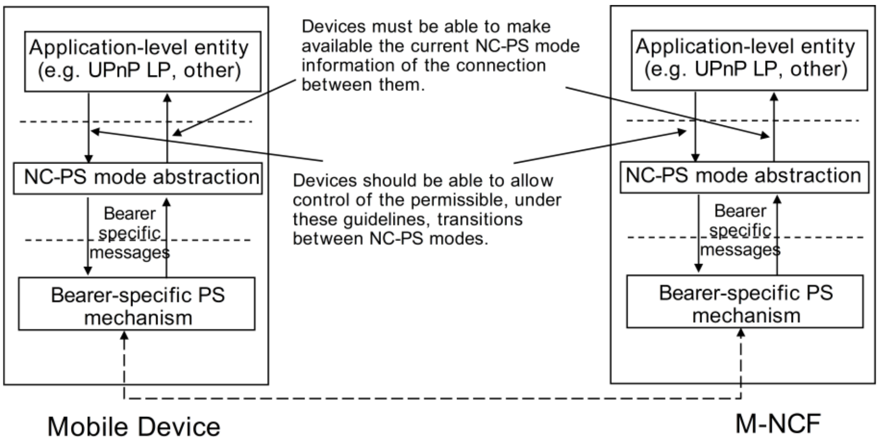


Figure G.1 — An illustration of the abstraction introduced by the NC-PS modes

The NC-PS guidelines do not introduce any new protocol for controlling transitions from one NC-PS mode to another; instead, they specify how the two ends use messages defined by the underlying bearer to trigger transitions and determine the current NC-PS mode between them. There is an inherent asymmetry between the mobile device and the M-NCF: the mobile device, as the power-constrained party, is the one that has more control in deciding when to put the link between them in a specific NC-PS mode. Even though the guidelines do not specify how often these transitions occur, entering a low-power NC-PS mode can have implications on higher-layer functionality of the mobile device (e.g. at the UPnP level), so manufacturers are cautioned to implement triggers that are invoked relatively infrequently (e.g. in the order of tens of seconds or more). Figure G.2 depicts the diagram of the allowed NC-PS mode transitions, which are further explained below:

- RAS -Transition from Active to Standby: The mobile device shall decide when this transition happens and the M-NCF shall agree. The trigger is left undefined, usually based on an inactivity timer or application level entities.
- TRAD -Transition from Active to Disconnected: Either the mobile device or the M-NCF may decide at any time to disconnect. The trigger is left undefined, usually based on a user indication or an application-level entity.

- TRSD -Transition from Standby to Disconnected: Either the mobile device or the M-NCF may decide at any time to disconnect. The trigger is left undefined, usually based on a user indication or an application-level entity.
- TRSA -Transition from Standby to Active: The mobile device shall decide when this transition happens and the M-NCF shall agree. The trigger is left undefined, usually based on the level of traffic encountered or an application-level entity.
- TRDA -Transition from Disconnected to Active: The mobile device should decide when this transition happens. The NCF may decide when this transition happens. The trigger is left undefined, usually when the user or an application-level entity perform a connect action.

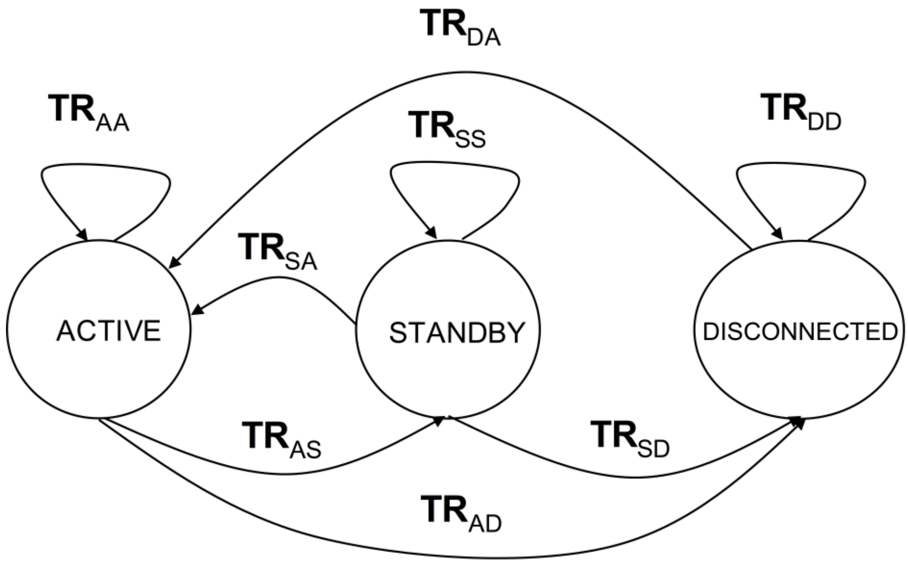


Figure G.2 — NC-PS Mode Transition Diagram

Finally, the M-NCF shall adapt its behavior to accommodate the NC-PS mode the connection is in, depending on whether it supports any of the two possible traffic-reduction operations specified in the NC guidelines: 7.2.5.5.9 (NC M-NCF: NC-PS Traffic Reduction Operations) and 7.2.5.5.6 (NC M-NCF: ARP proxying functionality). When the M-NCF performs UPnP filtering it prevents UPnP multicast messages from reaching the mobile device. This allows the mobile device to conserve energy since it has to spend less time communicating, but at the same time its UPnP state can become outdated. When the device moves out of Standby mode, it needs to re-establish its UPnP state through a UPnP SEARCH mechanism. On the other hand, when the M-NCF performs ARP proxying it stops forwarding ARP requests to the mobile device, while at the same time it assumes the responsibility of responding to ARP requests on behalf of the mobile device. This operation has no adverse effects on the mobile device. Furthermore, since ARP traffic is very frequent in Ethernet-based LANs, this operation can result in significant energy savings for the mobile devices. Upon determining that the connection has entered a different low-power NC-PS mode, the M-NCF shall adapt its behavior according to Table G.1.

Table G.1 — Dynamic Behavior of the M-NCF Depending on the Current NC-PS Mode

NC-PS Mode	M-NCF Behavior
Active	The M-NCF performs no operations on behalf of the mobile device.

NC-PS Mode	M-NCF Behavior
Standby	The M-NCF performs any, both or none of (a) ARP proxying (b) UPnP filtering on behalf of the mobile device.
Disconnected	The M-NCF performs no operations on behalf of the mobile device.

Annex H
(informative)

RTP Protocol Stack and SDP/RTSP/RTCP Parameters

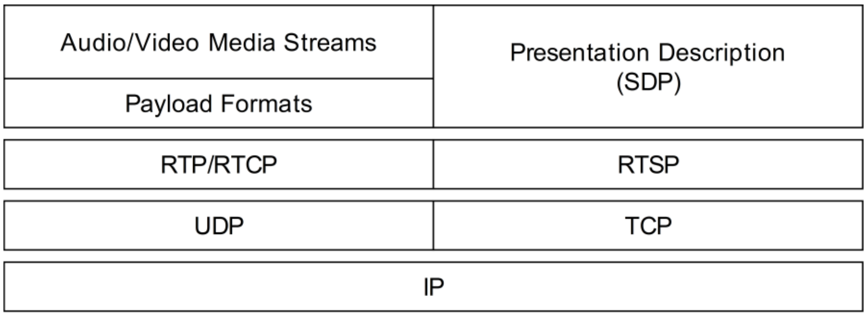


Figure H.1 — Overview of the protocol stack for RTP transport

SDP and RTSP Parameters

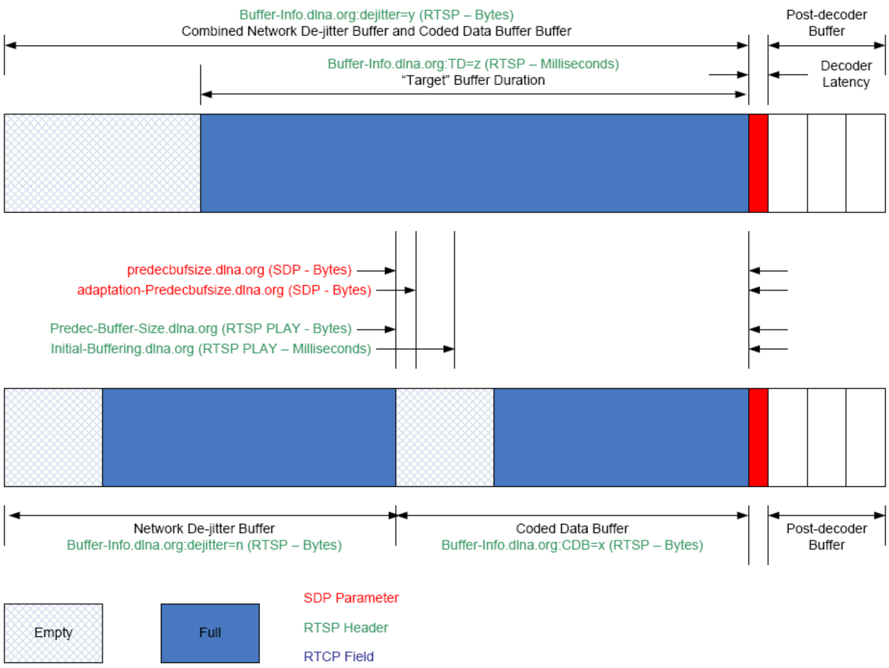


Figure H.2 — SDP and RTSP Parameters

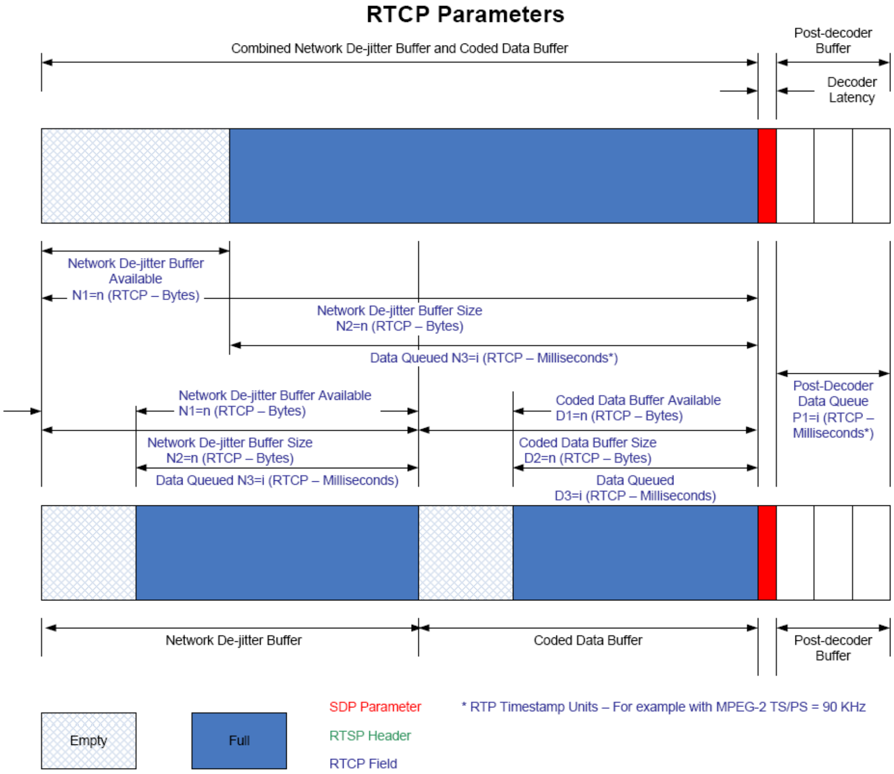


Figure H.3 — RTCP Parameters

Annex I
(informative)

Address Conflict Resolution in Auto-IP

I.1 Guidance

Autoconf or Zerconf address allocation may be used for DLNA devices in the absence of a DHCP address server. There are a number of different Internet Draft versions of AutoIP published ending finally in RFC 3927. The issue at hand is that there are subtle variations in the requirements from these different drafts. Ideally, all endpoints would implement the requirements of RFC 3927, providing for interoperability.

However, a most popularly implemented draft - draft-cheshire-ipv4-acd-03.txt - published 9th December 2002 defines the following options in behavior when resolving conflicts in addresses. The following text is an excerpt from this draft:

“At any time, if a host receives an ARP packet (request *or* reply) where the 'sender IP address' is (one of) the host's own IP address(es), but the 'sender hardware address' does not match any of the host's own interface addresses, then this is a conflicting ARP packet, indicating some other host also thinks it is validly using this address. To resolve the address conflict, a host shall respond to a conflicting ARP packet as described in either (a), (b) or (c) below:

(a) Upon receiving a conflicting ARP packet, a host MAY elect to immediately cease using the address, and signal an error to the configuring agent as described above, or

(b) If a host currently has active TCP connections or other reasons to prefer to keep the same IP address, and it has not seen any other conflicting ARP packets recently (for Ethernet, within the last ten seconds) then it MAY elect to attempt to defend its address. To defend its address, the host first records the time that the conflicting ARP packet was received, and then broadcasts one single ARP announcement, giving its own IP and hardware addresses. Having done this, the host can then continue to use the address normally without any further special action. However, if this is not the first conflicting ARP packet the host has seen, and the time recorded for the previous conflicting ARP packet is recent (within ten seconds for Ethernet) then the host SHALL immediately cease using this address and signal an error to the configuring agent as described above. This is necessary to ensure that two hosts do not get stuck in an endless loop with both hosts trying to defend the same address.

(c) If a host has been configured such that it should not give up its address under any circumstances (perhaps because it is the kind of device that needs to have a well-known stable IP address, such as a link's default router, or a DNS server) then it MAY elect to defend its address indefinitely. If such a host receives a conflicting ARP packet, then it should take appropriate steps to log useful information such as source Ethernet address from the ARP packet, and inform an administrator of the problem. The number of such notifications should be appropriately controlled to prevent an excessive number of error reports being generated. If the host has not seen any other conflicting ARP packets recently (for Ethernet, within the last ten seconds) then it SHALL record the time that the conflicting ARP packet was received, and then broadcast one single ARP announcement, giving its own IP and hardware addresses. Having done this, the host can then continue to use the address normally without any further special action. However, if this is not the first conflicting ARP packet the host has seen, and the time recorded for the previous conflicting ARP packet is recent (within ten seconds for Ethernet) then the host SHALL NOT send another defensive ARP announcement. This is necessary to ensure that two misconfigured hosts do not get stuck in an endless loop flooding the network with broadcast traffic while they both try to defend the same address.”

Annex J
(informative)

Wi-Fi Direct for DLNA

J.1 Wi-Fi Direct Introduction

WFA started certifying Wi-Fi CERTIFIED™ Wi-Fi Direct devices in 2010. Previous to Wi-Fi Direct, the focus of Wi-Fi, as it relates to DLNA, was a wireless home networking technology. With traditional Wi-Fi, an Access Point (AP) was required to allow communication within a home network. Wi-Fi Direct is different in that devices communicate directly amongst each other over Wi-Fi without an AP. Wi-Fi Direct is a device-to-device communication technology. Removing the dependency on the AP allows device-to-device communication for sharing, printing, showing and synchronizing in any location, with or without an AP.

Wi-Fi Direct is an extension to the Wi-Fi transport or link layer technology from a DLNA perspective. TCP/IP Networking operates over Wi-Fi Direct in the same manner as traditional Wi-Fi. Wi-Fi Direct Certification requires each Wi-Fi Direct device to support Wi-Fi Simple Config™ enrollee and Internal Registrar functionality, and also requires a DHCP server and DHCP client in order to provide devices with a proper IP address.

J.1.1 Terminology

According to the WFA Wi-Fi Direct Certification, a **P2P device** is capable of two roles:

- P2P Group Owner (GO) role: An “AP-like” capability that controls a Wi-Fi P2P Group and enables P2P Device connectivity;
- P2P Client role: A Wi-Fi P2P-compliant device that can connect to a P2P Group Owner

Once created, a **P2P Group** (see Figure J.1) can be comprised of both P2P devices and legacy devices. A legacy device can only be a client within a P2P Group. The created P2P Group can be classified as *temporary* (single-session) or *persistent* (multiple sessions) for which credentials are retained for subsequent sessions’ establishment.

P2P Group Owners can support optional features such as:

- Providing intra-BSS distribution enabling communication between members of the group;
- Supporting simultaneous (concurrent) connection with an infrastructure network and cross-connection to provide P2P Clients access to a simultaneous infrastructure connection

Figure J.1 — P2P Group

A P2P Group is uniquely identified by a **P2P Group ID**, composed of the P2P Device address of the GO, and an SSID that begins with the ASCII characters “DIRECT-“.

J.1.2 Group Formation

Figure J.2 illustrates the procedures by which two P2P devices form a new P2P Group.

The scan phase performed on all channels supported by the P2P device serves two main purposes:

- Discover P2P Devices that are currently member of an operational P2P Group
- Collect information about surrounding networks, and identify best potential operating channel(s) for establishing a new P2P Group.

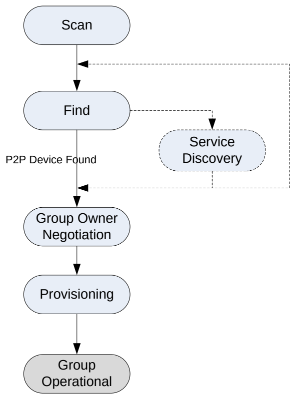


Figure J.2 — Group Formation Simplified Diagram

The **Find** phase is used for two searching P2P devices to find each other in view of (re-)establishing a P2P Group using the listen channel of one of the devices for initial signalling. During the Find phase, a searching P2P device alternates between search and listen state (see Figure J.3).

- In the **Search** state, the P2P device transmits one or more Probe Request frames on each of the ***Social*** Channels, i.e. channel 1, 6, and 11 in the 2.4 GHz band. The Probe Request contains both P2P and WSC attributes. To narrow its search, the Probe Request can contain one or more Requested Device Types or a P2P Device ID. In that case, only devices that match the request will send a Probe Response;
- In the **Listen** state, the P2P device tunes to its chosen *Listen* channel, one among the three social channels, and responds to received Probe Requests as required. The listen state duration is randomized to ensure that two searching devices will eventually find each other.

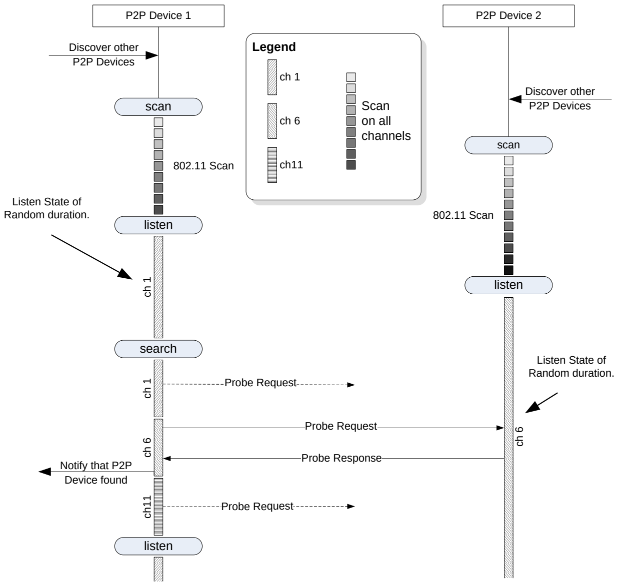


Figure J.3 — Device Discovery Procedure

Once a P2P Device has found another P2P Device, it can optionally invoke P2P *Service Discovery* to verify that both devices implement compatible services.

During the Group Owner negotiation phase, the two devices use a 3-way handshake to negotiate which one will take the role of P2P Group Owner based on user preference and/or capabilities, encoded as GO Intent value (0-15). Group Owner negotiation fails if both devices require to be the Group Owner (intent value= 15). Other group parameters are also negotiated, such as Group Operating Channel, Group duration (temporary or persistent), Group credentials, Group capability…etc.

Group Provisioning takes place after Group Negotiation, all information required to execute provisioning (e.g., PIN from a label or from the display, etc.) is obtained prior to P2P Group Formation. The new P2P Group Owner starts the P2P Group session using the credentials established during Group Owner Negotiation and then allows association by the Wi-Fi Direct device with which it is in Group Formation.

J.1.3 P2P Group Operation

Once a P2P Group has been formed, data is exchanged between the P2P Group Owner and each connected Client, using WPA2-Personal security with AES-CCMP as the encryption cipher. A P2P Group Owner can also provide intra-BSS distribution services between P2P Clients in its group.

Both devices can employ power savings techniques to address battery-operated devices requirements:

- A P2P Client uses standard mechanism for indicating that it is using power management and transiting from doze to awake state, with adapted mechanisms due to GO Power Saving or unavailability;

- A P2P GO uses Wi-Fi Direct specific mechanisms for power saving or time-sharing other activities. Such mechanism can be used as long as no legacy stations have joined the group. Indeed, a legacy device expects the GO to behave just like an AP and in particular, be always available. The P2P GO advertises its periods of availability/ unavailability in Beacon. These periods can be of type "one off" or "periodic" depending of traffic.

  P2P Clients can influence the use of P2P Power Save by submitting a P2P Presence Request when they have specific traffic requirement, such as when transmitting latency-sensitive data.

The duration that communication is unavailable due to power savings is on the order of 10's of milliseconds with little impact expected to higher layer protocols and applications.

In addition to arbitrating group access and data communication, the GO also provides *client discovery* services, by including device information and available services for each P2P Client associated to it, in Probe Responses returned to P2P Devices on receiving a Probe Request.

A P2P Group session ends when the GO leaves the group. A persistent Group (consisting of multiple sessions) ceases to exist when the GO deletes the stored credentials for that group.

J.1.4 Features that are optional in Wi-Fi Direct Certification

Additional procedures and operations are defined to enhance the basic ones:

J.1.4.1 Service Discovery

This procedure enables the advertisement of services supported by higher layer applications (e.g. UPnP, Bonjour) to other Wi-Fi Direct devices and prevents two devices from forming a new P2P group just to discover that they implement incompatible services. Service Discovery can be performed at any time (e.g. even before a connection is formed) with any other discovered Wi-Fi Direct device.

J.1.4.2 Persistent Group Re-invocation

One of the advantages of making a P2P Group persistent is that the group can be restarted without provisioning, thus eliminating the need for user intervention to repeat provisioning, e.g. entering a PIN. For example, a user can create a persistent group between its mobile phone (DMS) and TV set (DMR) so that each time it wishes to show/ watch a recorded video or synchronise both music stores, the P2P Group is restarted. To invoke a Persistent P2P Group, a P2P Client first discovers the P2P Group Owner, which may then camping for an *extended listening* period on a listen channel, and then completes a P2P Invitation exchange with the P2P Group Owner. Alternatively, a P2P Group Owner can invoke a Persistent P2P Group autonomously at any time (for example, in response to a request from a higher application layer).

J.1.4.3 Invitation Procedure

This procedure allows either a member of a currently operational P2P Group to request another P2P device to join its group, or a P2P device to re-invoke a persistent group. If a device supports persistent group, it also supports the invitation procedure.

J.1.4.4 Concurrent Operation

This is the capability for a P2P device to "simultaneously" join an infrastructure network and be a member, i.e. P2P GO or P2P Client, of a P2P Group. An example of a concurrent device is a laptop participating as a P2P Client and simultaneously using a WLAN connection to access the internet. Concurrent operation requires support for two distinct MAC entities – one for operation as a WLAN-STA and one for operating as a Wi-Fi Direct device, and at least two distinct interface addresses.

J.1.4.5 Intra-BSS Distribution (Required for DLNA Certification)

In a P2P Group, the GO supports Intra-BSS distribution (bridging) service between all connected Clients in the P2P Group. Without such feature, communication can only take place on a one-to-one basis, between the GO and each connected client. If two clients want to communicate, they have to form another group, possibly leaving the original group unless they support multiple interfaces. Intra-BSS Distribution within the P2P Group takes place at layer 2.

J.1.4.6 Cross-connection

This is the capability for a concurrent P2P GO to route traffic from the P2P Group to the infrastructure network, and vice-versa. Cross-connection between a WLAN and a P2P Group uses mechanisms above layer 2. A P2P Client cannot cross-connect.

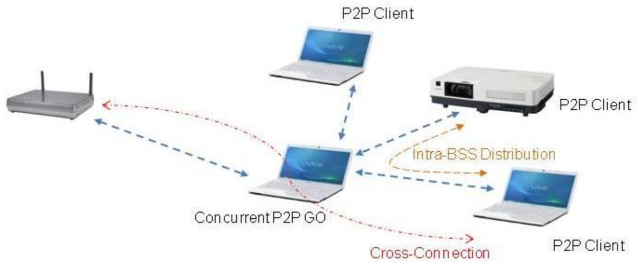


Figure 4 — Intra-BSS distribution & Cross-connection

J.2 Wi-Fi Direct with System Usages

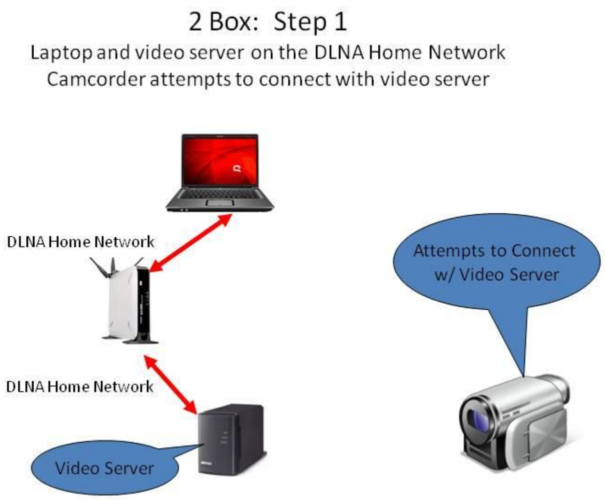
Because Wi-Fi Direct operates just like any other IP-based network, Wi-Fi Direct is a DLNA network. DLNA 2 Box and 3 Box System Usages work the same over a DLNA Wi-Fi Direct Network and a DLNA Home Network with an AP when devices remain present. However, Wi-Fi Direct has an ephemeral quality intended to accomplish a specific user task. The user initiates setup of the Wi-Fi Direct link, performs an action, and then tears down the link. Most of DLNA is designed around long term network connectivity, which is unlike the temporary design of Wi-Fi Direct. If long term network connectivity is required, then traditional Wi-Fi communication through an AP is expected.

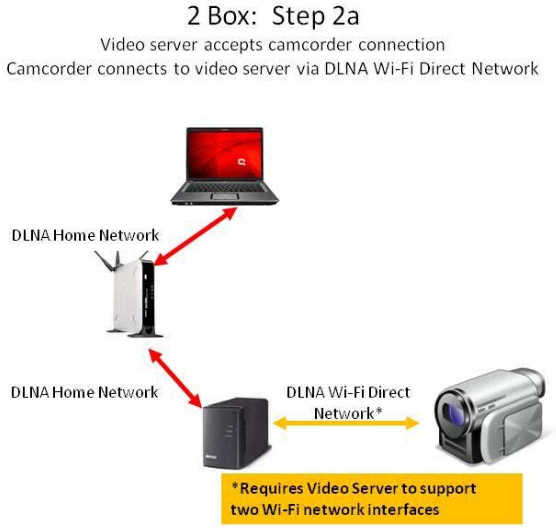
J.2.1 and J.2.2 examines how an ephemeral Wi-Fi Direct connection can affect the assumptions of long term network connectivity within the DLNA Two-box and Three-box System Usages. To be more specific, if a user starts a Wi-Fi Direct task after devices are on a DLNA home network, then applications assuming the devices are still on the home network might not successfully operate because IP addresses and IP subnets might change as the Wi-Fi Direct link activates. DLNA networks today have a similar problem if they are reconfigured by changing cables or installing new networking gear. In both the Two-box and Three-box cases, implementation decisions can be made to mitigate the problem.

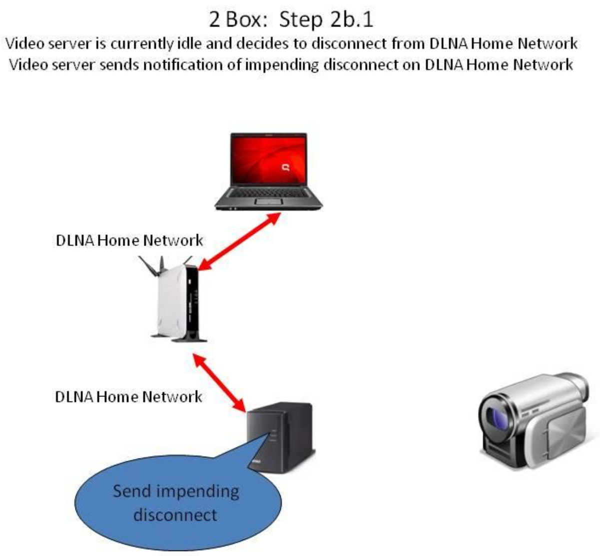
J.2.1 and J.2.2 describe specific scenarios of the problems that can occur and implementation methods to mitigate those problems.

J.2.1 Two-Box System Usage

J.2.1 examines the two-box system usage. The Wi-Fi Direct user action in this example is to connect a camcorder to a video server already on the DLNA home network. The implementation choices and mitigation methods for the two-box system usage are illustrated in Steps 2a and Steps 2b in Figure J.5.




2 Box: Step 2a
Video server accepts camcorder connection
Camcorder connects to video server via DLNA Wi-Fi Direct Network
DLNA Home Network
DLNA Home Network
DLNA Wi-Fi Direct Network*
*Requires Video Server to support two Wi-Fi network interfaces


2 Box: Step 2b.1
Video server is currently idle and decides to disconnect from DLNA Home Network
Video server sends notification of impending disconnect on DLNA Home Network
DLNA Home Network
DLNA Home Network
Send impending disconnect

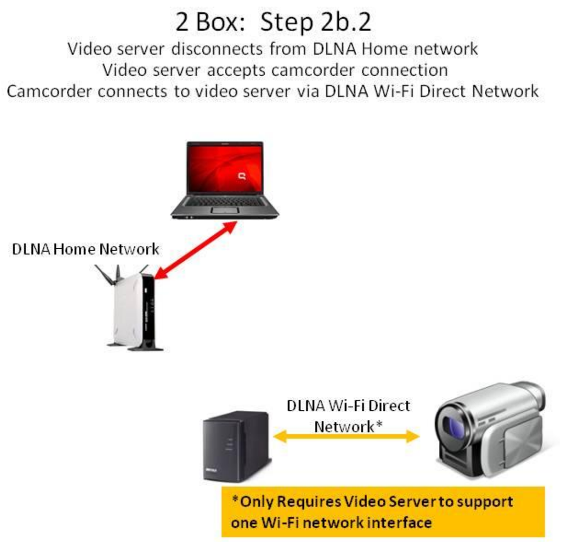


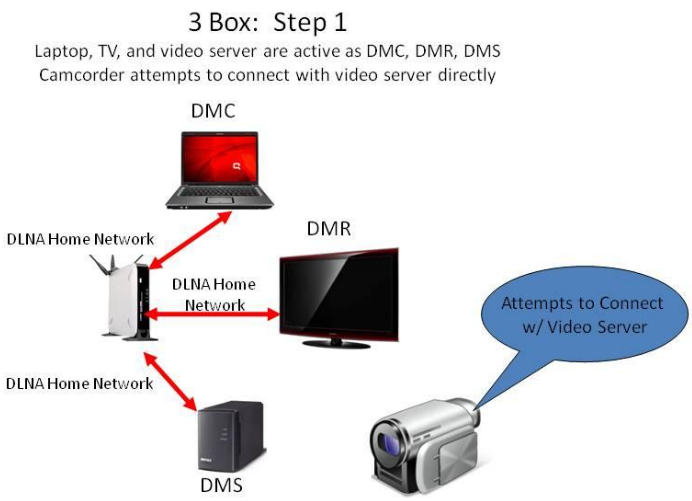
Figure J.5 — 2-Box System Usage Example

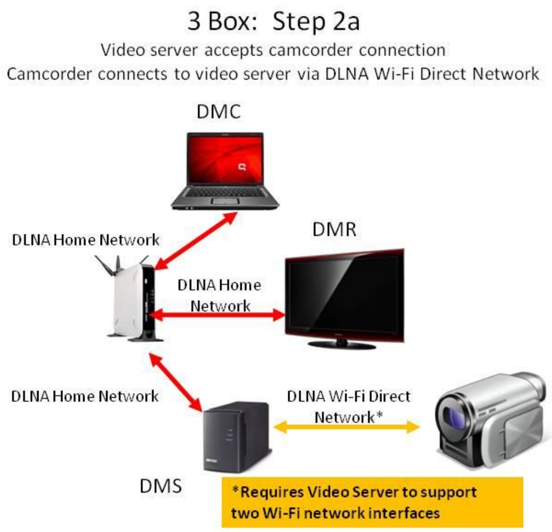
The video server implementation that allows for multiple network interfaces has no interruption of service from the DLNA perspective (Step 2a in Figure J.5).

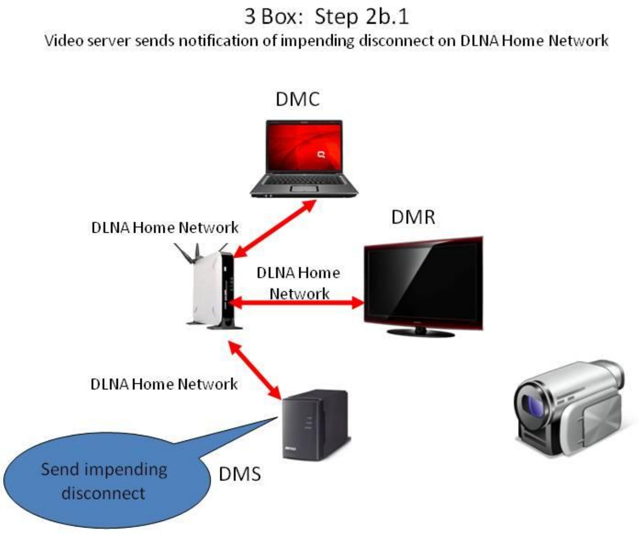
The video server implementation that does not allow for multiple network interfaces can have an interruption of service between the laptop and video server (Step 2b in Figure J.5). Step 2b in Figure J.5 interruption of service can be mitigated by the video server deciding to disconnect only if the video server is idle or sending an impending disconnect, and then automatically reconnecting back to the DLNA Home Network once the camcorder and video server finish synchronization. The video server implementation has the option to not accept the connection from the camcorder.

J.2.2 Three-Box System Usage

J.2.2 examines the three-box system usage. The Wi-Fi Direct user action in this example is to connect a camcorder to a video server already on the DLNA home network acting as a DMS. The implementation choices and mitigation methods for the three-box system usage are illustrated in Steps 2a and Steps 2b in Figure J.6.


3 Box: Step 1
Laptop, TV, and video server are active as DMC, DMR, DMS
Camcorder attempts to connect with video server directly
DMC
DLNA Home Network
DMR
DLNA Home Network
Attempts to Connect w/ Video Server
DLNA Home Network
DMS


3 Box: Step 2a
Video server accepts camcorder connection
Camcorder connects to video server via DLNA Wi-Fi Direct Network
DMC
DLNA Home Network
DMR
DLNA Home Network
DLNA Home Network
DLNA Wi-Fi Direct Network*
DMS
*Requires Video Server to support two Wi-Fi network interfaces



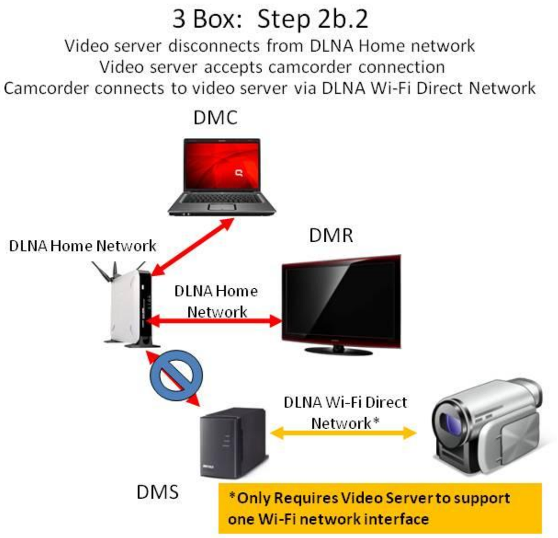


Figure J.6 — 3-Box System Usage Example

The video server implementation that allows for multiple network interfaces has no interruption of service from the DLNA perspective (Step 2a in Figure J.6).

The video server implementation that does not allow for multiple network interfaces can have an interruption of service between the laptop and video server (Step 2b in Figure J.6). Step 2b in Figure J.6 interruption of service can be mitigated by the video server sending an impending disconnect, and then automatically reconnecting back to the DLNA Home Network once the camcorder and video server finish synchronization.

Annex K
(informative)

EPG Theory of Operation

K.1 Goal

The DLNA EPG device option allows a DMS (UPnP AV Media Server) to provide Electronic Program Guide data within the home. A Client in the home can access this data to present an Electronic Program Guide to the user. The client is an EPG Controller (UPnP AV control point) that accesses the Media Server's CDS by issuing FreeFormQuery search commands to the server to obtain the program information. EPG data can be provided for channelized content that is scheduled and associated with a specific delivery channel such as a tuner. Non Channelized content such as Video On Demand (VOD) can also be described.

K.2 Usage Scenarios

A typical usage scenario of the DLNA EPG device option is a set-top box that acts as a DMS. This set-top box obtains the EPG data from an external source, and makes it available on the home network. For example, this set-top box would be connected to a TV in the living room. The EPG is available on the TV the set-top box is attached to. However, using another DLNA TV with integrated EPG Controller, the user is able to view the EPG in other rooms as well. Instead of a DLNA TV, a light-weight set-top box can be used to access the EPG. When the set-top box that provides the EPG contains storage, it can expose its contents through the DMS (UPnP AV Media Server), allowing to user to view recorded programs on a client TV. The EPG Extended Tuner enables the user to watch live content directly from the tuner of the set-top box. Further, if the Scheduled Recording device option is added to the DMS, and a compatible EPG Controller is available in a client, the client can provide functionality to the user such as selecting live or recorded content, viewing the EPG, and scheduling recordings.

Another scenario that is enabled by the EPG device option is the use of an EPG data only server. The user subscribes to a service which offers more extensive EPG data for a certain set of channels. The application would retrieve the EPG data offered by such a service and make it available to any DLNA EPG control point in the home. The advantage is that the clients make use of the DLNA standard while the EPG service can use a proprietary way of providing the fuller EPG data.

K.3 The Model

K.3.1 EPG Data

The fundamental mechanism in the DLNA EPG Server device option is to provide a small mandatory set of properties. The mandatory set allows an EPG control point to render a basic EPG. Today many services exist that offer a so-called “rich” EPG and these services differ in what type of metadata they offer to the user. A “rich” EPG service could, for example, provide access to similar programs, provide options to buy a program, cluster channels, access additional information on the internet, add advertisements, etc. Since it would be difficult to define new UPnP properties for all this metadata, the concept of “foreign metadata” is introduced. Using the foreign metadata approach, any XML based metadata can be added to a CDS object without requiring UPnP CDS changes (an EPGItem in the case of EPG data).

A DMS (UPnP AV MediaServer) that implements the EPG device option will assign values to the mandatory properties of EPG elements. The guidelines describe what type of information will be assigned to each mandatory property. A detailed description for each property is provided for the following standards; OpenEPG, TV-Anytime, and DVB-SI.

For example, consider a DMS implementing the EPG device option that uses an OpenEPG based EPG description as its source. In this case, the guidelines define which OpenEPG data elements will be assigned to which CDS properties. If a certain OpenEPG data element is not present, the guidelines define the information which will be assigned as an alternative. Since the OpenEPG input format can provide much more information than the small set of mandatory properties, the additional data can be added to an EPGItem as foreign metadata. A client that is able to parse OpenEPG data elements can use the OpenEPG based foreign meta-data to provide a richer EPG. A client which cannot parse the OpenEPG data can still show a basic EPG.

K.3.2 FreeFormQuery

The EPG device option defines how the EPG data is presented as a set of EPGItem objects in the CDS. Typically an EPG control point would search the CDS to obtain relevant EPGItems. The defined mapping of EPG data elements into CDS properties allows the UPnP CDS search action to have access to the basic EPG information. Access to the additional information stored in foreign metadata uses FreeFormQuery. The FreeFormQuery search mechanism allows a search to be specified using the XQuery language. Typically an XQuery would be used to constrain the amount of EPG data returned to the querier. For example, a query could be constructed which restricts the EPG data to a limited set of channels and a time window to build an EPG grid screen.

The XQuery Language is a complex language that allows searching (and modifying) any XML based document. Using a complete XQuery engine, a DLNA EPG Server would need to represent the EPGItems in the CDS as an XML document, regardless of how the information is actually stored. The XQuery engine uses this XML document as the source against which the query will be executed. For a small embedded device it can be preferable to store the EPGItems in a local database. Since executing an arbitrary XQuery on a database in a compact and efficient way is difficult or impossible, two levels of XQuery subsets have been defined. The first XQuery subset, which is specified in 7.4.6.7.5, defines the bare minimum that needs to be implemented in any DLNA EPG Server. This subset can be used on simple servers and is sufficient for common EPG searches. The second subset, specified in 7.4.6.7.6, enables more complex searches while still being able to map an XQuery to an internal database. The first subset is a subset of the second subset. Using this approach allows data from an EPG source to be efficiently stored in a local database and each XQuery is parsed and translated to a local database query. Through the CDS:FreeFormCapabilities action, a UPnP AV Media Server can indicate which properties it emits in an XQuery. A small list of mandatory properties is defined in the guidelines. A server can chose to support searching of foreign metadata properties as well. Using the CDS:FreeFormCapabilities action an EPG control point can determine, for example, if a server supports an XQuery that returns a list of movies from a certain director in a certain year using TV-Anytime based foreign-metadata. Likewise, an EPG control point can determine if a server supports an XQuery that returns a list of genres or other features that are specified in other foreign-metadata formats.

K.3.3 Channel Lineup

To indicate the channels available in an EPG, the DMS exposes a list of channels by implementing the DLNA Extended Tuner [Integration note: add reference to section]. The DLNA Extended Tuner defines Channel Lineup Containers, which are CDS containers that contain a list of channels in the form of videoBroadcast item objects or audioBroadcast item objects. To determine which channels are available an EPG control point can search or browse the Channel Lineup Container. Each videoBroadcast item can have a ChannelName that is used to represent the channel to the user (in contrast, the dc:title property typically represents the name of the currently available broadcast item). Additionally, each videoBroadcast or audioBroadcast item will have a upnp:channelID property. The upnp:channelID property is used as the main mechanism to link EPGItems to Channels. To find EPGItems with a particular channel name, the EPG Controller first queries the tuner object for the desired channel name and then uses the resultant upnp:channelID property value to search the EPGItem list. Searching for all EPGItems with a certain upnp:channelID property value results in a list of all known programs for the desired channel. Note that a device does not need to have a physical tuner in order to implement the tuner feature. If no physical tuner is available in the device, or the vendor does not want to provide direct access to the tuner's live content from other networked devices, the <res> elements simply do not contain a URL. In this case, the tuner feature merely provides a way to communicate the channel line-up.

K.3.4 Channel Ordering

The preferred method for determining the order of the channels is to use the ordering of the videoBroadcast items (or audioBroadcast items) in the Channel Lineup Container. When a physical tuner is exposed through the CDS, each videoBroadcast item will contain a <res> element. Reading from the URL will produce the video stream corresponding to the selected channel. When the user presses the channel up button on the remote control, the control point uses the next item in the Channel Lineup Container.

To distinguish between the cases described above the upnp:channelID property has a @type attribute. In case of a DVB based system, the value of the channelID@type attribute will be "SI". The upnp:channelID property value will contain a DVB triplet consisting of "<Network ID>,<Transport Stream ID>,<Service ID>". In other systems where channels are indicated using a major and a minor number (i.e. used in terrestrial digital broadcast systems), the value of the upnp:channelID@type property will be "DIGITAL". The upnp:channelID property will contain the major and minor number separated by a comma. Finally, in systems where the channel numbering consists of integers only, the value of the upnp:channelID@type property will be "ANALOG". The upnp:channelID property contains the channel number.

K.3.5 channelID@distriNetworkID

The upnp:channelID@distriNetworkID attribute value will contain a value indicating the network from which a program is distributed and allows channels to be queried by the distribution source. Note that the upnp:channelID@distriNetworkID attribute was defined in AVv3 as an additional channelID qualifier.

K.3.6 Advanced Lineup

Digital broadcast systems have the capability of supporting a number of channels in excess of a thousand. In such systems, channels are often grouped. This can be accomplished by adding containers to the DLNA Extended Tuner. Each container will contain a set of channels.

Since the DLNA Extended Tuner is intended to expose a channel line up through the CDS, rather than expose a physical tuner, the number of Channel Lineup Containers is not necessarily tied to the number of physical tuners in a system. For example, a device could have two physical tuners and expose just one Channel Lineup Container representing a single line up. It is also possible to have a system with no physical tuner, and one or more different line ups. The latter could be used in the scenario where a PC based EPG Server offers rich EPG descriptions for other devices.

It is also possible to expose multiple views of the channel lineup of a tuner through the use of Favorites and Presets Containers. For example, while the Channel Lineup Container contains the list of all channels, a Favorites or Presets Container would show only the channels which the user subscribes to. Other Favorites or Presets Containers can also be defined to allow users to compile a list of favorite channels.

K.4 Implementation Considerations

K.4.1 General

To clarify the model above, K.4 explains the implementation aspects from the perspective of an EPG control point.

K.4.2 Discovering Features and Capabilities

An EPG control point can invoke the CDS:GetFeatureList action to obtain information on the EPG, the Tuner(s), foreign-metadata, and FreeFormQuery.

K.4.3 Discovering EPG Servers

An EPG control point can discover EPG Servers, by invoking the CDS:GetFeatureList action on a particular server. If the DMS implements the EPG device option, the result of the GetFeatureList action will contain a Feature-element with its name attribute set to “EPG” or “DLNA.ORG_EPGDataOnly”.

K.4.4 Discovering Tuners

If a UPnP AV Media Server implements the EPG Server device option, then it will also expose at least one DLNA Extended Tuner. This tuner is used to provide the channel line-up. The tuner feature-element returned as a result of the CDS:GetFeatureList action will contain one or more objectIDs. These objectIDs identify the tuner containers.

K.4.5 Determining FreeFormQuery Capabilities

Additionally the CDS:GetFeatureList action will indicate on which containers the CDS:FreeFromQuery is supported. This is indicated through a list of ObjectId elements.

An EPG Server always supports the CDS:FreeFormQuery action for all EPG containers or an ancestor container of each EPG container.

For each ObjectId listed as part of the FFQ <Feature> element, the extent of support for the XQuery language is denoted by the level attribute. If the objectID@level attribute is set to “0” this means that the full XQuery language is supported for this container. If the objectID@level attribute is set to “DLNA_EPG” this means that only the minimal subset of XQuery is supported. If the objectID@level attribute is set to “DLNA_EPG_EXPANDED” the extended XQuery subset is supported.

K.4.6 GetFeatureList Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Features
   xmlns="urn:schemas-upnp-org:av:avs"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:av:avs
http://www.upnp.org/schemas/av/avs.xsd">
   <Feature name=”EPG” version=”1”>
      <objectIDs>
         xxx,yyy,zzz
      </objectIDs>
   </Feature>
   Feature name=”TUNER” version=”1”>
      <objectIDs>
         T1
      </objectIDs>
   </Feature>
   <Feature name=”FFQ” version=”1”>
      <objectID level =”DLNA_EPG_EXPANDED”>
```

```
            yyy
        </objectID>
        <objectID level =”DLNA_EPG_EXPANDED”>
            yyy
        </objected>
        <objected level =”DLNA_EPG”>
            zzz
        </objectID>
    </Feature>
    <Feature name="FOREIGN_METADATA" version="1">
        <type id=”openepg.org_v1" provider="dish.org"></type>
        <type id="tv-anytime.org" provider="tribune.org"></type>
    </Feature>
</Features>
```

K.4.7 Determining FreeFormQuery Capabilities

To discover which properties can be used in CDS:FreeFromQuery action, a control point invokes the CDS:GetFreeFormCapabilities action. This action returns a <propertyList> element containing <propertyName>-elements. The element lists the names of the properties that can be used in the XQuery. Optionally the CDS:GetFreeFormCapabilities action will return a <searchOnlyPropertyList> element. This list contains <propertyName> elements that can not be used in the order-by clause.

K.4.8 Retrieving a Channel Lineup

To obtain a channel line up the EPG Server control point issues a browse or search starting at the root Channel Lineup Container. If the server implements the “DLNA_EPG_Expanded” XQuery subset or the full XQuery language it is possible to search for videoBroadcast items in a Channel Lineup Container. If the CDS:GetFreeFormCapabilities action indicates that it supports the use the ChannelID property and the channelID@distriNetworkID property in the XQuery, the following example shows a way to obtain the first 10 channels. (In a system that makes use of DVB, the “order by” clause could list the channelNr for sorting.)

FreeFromQuery (xquery, channel lineup container id)

```
<DIDL-Lite
xmlns=“urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/”
xmlns:upnp=“urn:schemas-upnp-org:metadata-1-0/upnp/”
xmlns:dc=“http://purl.org/dc/elements/1.1/”>
{
    (
        for $x in DIDL-Lite//item[fn:starts-with(upnp:class,
       “object.item.VideoBroadcast”) and fn:not(fn:exists(@refID))]
       order by $x/upnp:channelID/@distriNetworkID ascending,
       $item/upnp:channelID ascending
        return $x
    )
    [fn:position()= (1 to 10)]
}
</DIDL-Lite>
```

If the DLNA_EPG query subset is supported, it is not possible to search for video broadcast items. In that case, the EPG Server control point can revert to the CDS:Browse action to obtain the channel line up.

K.4.9 Obtaining an EPG Grid

(These are examples of EPG_EXPANDED queries)

Searches on EPG data should pass an ObjectID (see 7.4.6.2) to restrict the search to the EPG tree only.

Retreiving EpgItem(s) constrained to specific range of channels and times:

```
<DIDL-Lite
xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/"
xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/"
xmlns:dc="http://purl.org/dc/elements/1.1/">
{
    (
       for $item in DIDL-Lite//item[fn:starts-with(upnp:class,
"object.item.epgItem") and fn:not(fn:exists(@refID))]
       where $item/upnp:scheduledStartTime >= "2008-08-10T14:30:00" and
$item/upnp:scheduledEndTime < "2008-08-10T13:00:00" and
       $item/upnp:channelID/@distriNetworkID = "example-tv" and
       ($item/upnp:channelID = "201" OR $item/upnp:channelID = "202"
       OR $item/upnp:channelID = "203")
       order by $item/upnp:scheduledStartTime ascending
       return <item>{$item/@id, $item/dc:title, $item/scheduledStartTime,
$item/scheduledEndTime }</item>
    )
    [fn:position()= (1 to 10)]
}
</DIDL-Lite>
```

Retrieving longDescription for 1 EpgItem:

```
<DIDL-Lite
xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/"
xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/"
xmlns:dc="http://purl.org/dc/elements/1.1/">
{
    (
       for $item in DIDL-Lite//item[fn:starts-with(upnp:class,
"object.item.epgItem") and fn:not(fn:exists(@refID))]
       where $item/@id = "current epg item"
       return <item>{$item/@id, $item/upnp:longDescription }</item>
    )
    [fn:position()= (1 to 10)]
}
</DIDL-Lite>
```

Alternative for retrieving longDescription

```
<DIDL-Lite
xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/"
xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/"
xmlns:dc="http://purl.org/dc/elements/1.1/"> {
    (
       for $item in DIDL-Lite//item
       where $item/@id = "current epg item"
       return <descr>{$item/@id, $item/upnp:longDescription }</descr>
    )
}
</DIDL-Lite>
```

Retreiving EpgItem(s) by Keyword search on all channels: (search for TV programs containing keyword in title or description in the coming two days)

```
<DIDL-Lite
xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/"
xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/"
xmlns:dc="http://purl.org/dc/elements/1.1/">
{
    (
       for $item in DIDL-Lite//item[fn:starts-with(upnp:class,
"object.item.epgItem") and fn:not(fn:exists(@refID))]
```

```
       where $item/upnp:scheduledStartTime >= “2008-08-10T00:00:00” and
$item/upnp:scheduledEndTime < “2008-08-12T00:00:00”
       (
       fn:contains($item/dc:title,”keyword”) or
fn:contains($item/upnp:longDescription,”keyword”) order by
$item/upnp:scheduledStartTime ascending
       return <item>{$item/@id, $item/dc:title, $item/upnp:longDescription}</item>
    )
    [fn:position()= (5 to 10)]
}
</DIDL-Lite>
```

Retreiving EpgItem(s) by Keyword search on a set of channels:

```
<DIDL-Lite
xmlns=“urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/”
xmlns:upnp=“urn:schemas-upnp-org:metadata-1-0/upnp/”
xmlns:dc=“http://purl.org/dc/elements/1.1/”>
{
    (
       for $i in DIDL-Lite//item[fn:starts-with(upnp:class,  “object.item.epgItem”)
and fn:not(fn:exists(@refID))]
       where $i/upnp:scheduledStartTime >= “2008-08-10T00:00:00” and
$item/upnp:scheduledEndTime < “2008-08-12T00:00:00” and
       $i/upnp:channelID/@distriNetworkID  = “example-tv“
       AND
       ($i/upnp:channelID = “201“ OR $i/upnp:channelID = “202“ OR $i/upnp:channelID
= “203“)
       AND fn:contains($i/dc:title,”keyword”) or
fn:contains($i/upnp:longDescription,”keyword”)
       order by $i/upnp:scheduledStartTime ascending
       return <item>{$i/@id, $i/dc:title, $i/scheduledStartTime, $i/scheduledEndTime
}</item>
    )
    [fn:position()= (1 to 10)]
}
</DIDL-Lite>
```

Retreiving EpgItem(s) by current and next program for a channel.

```
<DIDL-Lite
xmlns=“urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/”
xmlns:upnp=“urn:schemas-upnp-org:metadata-1-0/upnp/”
xmlns:dc=“http://purl.org/dc/elements/1.1/”>
{
    (
       for $item in DIDL-Lite//item[fn:starts-with(upnp:class,
“object.item.epgItem”) and fn:not(fn:exists(@refID))]
       where $item/upnp:scheduledStartTime >=  “time 1” and
$item/upnp:scheduledEndTime < “time 2” and
       $item/upnp:channelID/@distriNetworkID = “example-tv“ and $item/upnp:channelID
= “201“
       order by $item/upnp:scheduledStartTime ascending
       return <item>{$item/dc:title, $item/scheduledStartTime,
$item/scheduledEndTime }</item>
    )
    [fn:position()= (1 to 2)]
}
</DIDL-Lite>
```

- Time 1 can be the current time
- Time 2 is sufficiently in the future. It cannot be omitted in the EPG subset (It can be omitted in EPG_Expanded).

Annex L
(Normative)

Rating Systems

Annex L lays out all the currently recognized rating systems and their attributes. These are summarized in Table L.1. The columns labeled “domain”, “valid ratings”, “age equivalence” and “valid advice” are all normative fields, and their use is prescribed by the guidelines in 7.4.6.5.6. All other columns are informative.

The column labeled “rating descriptions” is intended as an informational field, to be interpreted as follows:

- all audiences – this rating indicates that the media is generally unrestricted and appropriate for all ages and groups
- X or over – this rating indicates that the media shall only be viewed by persons over the age of “X”
- X or over, supervised – this rating indicates that the media shall only be viewed by persons over the age of “X”, but may be viewed by persons under the age of “X” with parental or adult supervision
- parental guidance – this rating indicates that the media may be viewed by persons of any age, but parental or adult supervision may be recommended for children
- exempt – this rating indicates that the media is exempt from classification, as may commonly be the case for sports media or documentary films
- banned –this rating indicates that the media has been banned in this country, and is not appropriate for viewing by persons of any age or group
- sexual – this item contains themes of a sexual nature
- graphic – this item contains graphic violence
- special – this content has a special rating, where special is defined by the local government or domain
- children – this rating indicates that the media may be viewed by persons of any age, and is noted to be especially friendly or informative to young children
- young adults – this content is appropriate for “young adults”, where the age range for young adults is not defined
- professional use – this content is intended for professional use
- adult – this content is only appropriate for “adults”, where the age range for adults is not defined
- under X – friendly to all ages, but especially intended for children under the age of X
- film only – in some domains where ratings apply to both film and TV, this rating only applies to movies
- restricted – may only be shown under certain (government specific) restrictions
- not for public viewing – these films may only be viewed in private residences
- rating pending – this content has not yet been rated, and is currently under review

The column labeled “age equivalence” is intended to be used programmatically to limit the viewing of programs. Note that this field is not guaranteed to be accurate, or reflect local laws where the EPG is being used or deployed, but is simply meant to be a best effort due to the lack of a global rating system. This column should be interpreted as follows:

- X – this content should not be viewed by persons under the age of X

- XPG – this content should be freely available to persons that are of the age X or over, and may be viewed by persons under the age of X with parental or adult supervision
- X-YPG – this content should not be viewed by persons under the age of X, and may be viewed freely by persons over the age of Y; persons that are at least X and younger than Y may view the content with parental or adult supervision
- <all ages valid> – this domain has age ratings for any integer that is equal to or larger than zero

Note that due to the way that various countries rate their films, the ratings from one country may not be transferrable to another country, despite the fact that some countries use common notations for their rating systems. As a hypothetical example, one country could allow limited nudity and adult language in a film that is rated for "all audiences", while another country may have an "all audiences" rating that does not allow for any nudity and language. It would be especially problematic if the second country banned nudity altogether, and an EPG developer allowed a film from the first country to be shown to children. The same applies to the age ratings of various counties, where moral and ethical judgments vary of which age group should be exposed to various adult themes.

Unfortunately, there is no way of translating from one parental control system to another, so the task of figuring out which film can be shown to which age group in which country is something that is left to the judgment of the system implementer.

Table L.1 — Rating Sysytems

Authority / Locale	Media Type	Domain	Valid Ratings	Rating Description	Age Equivalence	Valid Advice	Advice Description
INCAA / Argentina	Film	incaa.gov.ar	ATP	all audiences	0		
			13	13 or over	13		
			16	16 or over	16		
			18	18 or over	18		
			X	explicit			
			E	exempt			
ACMA / Australia	TV	acma.gov.au	P	pre-school		A	adult
			C	children		V	violence
			G	all audiences	0	L	language
			PG	parental guidance		S	sex
			M	mature		H	horror
			MA15+	15 or over	15	D	drugs
			AV15+	15 or over	15	N	nudity
						SN	supernatural
						M	medical
						W	war
						B	colorful behavior
Classification Review Board / Australia	Film	classification.gov.au	E	exempt			
			G	all audiences			
			PG	parental guidance			
			M	mature			
			MA15	15 or over	15		
			R18+	18 or over	18		
			X18+	18 or over, sexual	18		
			RC	banned			
BMUKK / Austria	Film	bmukk.gv.at	Altersstufen	all audiences	0		

Authority / Locale	Media Type	Domain	Valid Ratings	Rating Description	Age Equivalence	Valid Advice	Advice Description
			6	6 or over	6		
			10	10 or over	10		
			12	12 or over	12		
			14	14 or over	14		
			16	16 or over	16		
			E	exempt			
Brazil	Film & TV	mj.gov.br	ER	children	0		
			L	all audiences	0		
			10	10 or over	10		
			12	12 or over	12		
			14	14 or over	14		
			16	16 or over	16		
			18	18 or over	18		
			E	exempt			
Canada	TV (English)	cbsc.ca/english	E	exempt			
			C	children, under 8	0		
			C8	children, 8 or over	8		
			G	all audiences	0		
			PG	parental guidance			
			14+	14 or over	14		
			18+	18 or over	18		
Canada	TV (French)	cbsc.ca/french	E	exemptées			
			G	général	0		
			8 ans+	Général-Déconseillé aux jeunes enfants	8		
			13 ans+	Cette émission peut ne pas convenir aux enfants de moins de 13 ans	13		
			16 ans+	Cette émission ne convient pas aux moins de 16 ans	16		
			18 ans+	Cette émission est réservée aux adultes	18		
Chile	TV	www.anatel.cl	I	children			
			17	children, 7 or over	7		
			I12	children, 12 or over	12		
			F	all audiences	0		
			R	adult supervision			
			A	adult	18		
Chile	Film	filmnacional.cl	TE	all audiences	0	18/S	sex
			14	14 or over	14	18/V	violence
			18	18 or over	18		

Authority / Locale	Media Type	Domain	Valid Ratings	Rating Description	Age Equivalence	Valid Advice	Advice Description
			E	exempt			
Columbia	Film	mincultura.gov.co	T 7 12 16 18 X Banned E	all audiences 7 or over 12 or over 16 or over 18 or over sexual banned exempt	0 7 12 16 18		
Denmark	TV	.dk_UNOFFICIAL	Green Yellow Red	all audiences parental guidance adult	0	No governing body. These terms are common usage.	
Denmark	Film	medieraadet.dk	A 7 11 15 E	all audiences 7 or over 11 or over 15 or over exempt	0 7 11 15		
European Union / PEGI	Games	pegi.info	3+ 7+ 12+ 16+ 18+ 4+ 6+	3 or over 7 or over 12 or over 16 or over 18 or over 4 or over (Portugal) 6 or over (Portugal)	3 7 12 16 18 4 6	Bad Language Descrimination Drugs Fear Sex Violence Gambling	
Finland	Film	vet.fi	K-3 K-7 K-11 K-13 K-15 K-18 K-E	3 or over 7 or over 11 or over 13 or over 15 or over 18 or over exempt	3 7 11 13 15 18		
France	TV	csa.fr	-10 -12 -16 -18	10 or over 12 or over 16 or over 18 or over	10 12 16 18		
France	Film	culture.gouv.fr	U -12 -16 -18 -E	all audiences 12 or over 16 or over 18 or over Exempt	0 12 16 18		

Authority / Locale	Media Type	Domain	Valid Ratings	Rating Description	Age Equivalence	Valid Advice	Advice Description
Germany	Film	spio.de	FSK 0 FSK 6 FSK 12 FSK 16 FSK 18	all audiences 6 or over 12 or over 16 or over 18 or over	0 6 12 16 18		
Germany	Games	usk.de	ohne ab 6 ab 12 ab 16 ab 18	all audiences 6 or over 12 or over 16 or over 18 or over	0 6 12 16 18		
Hong Kong	Film	tela.gov.hk	I IIA IIB III IV	all audiences children, supervised young adults, supervised 18 or over exempt	0 18		
Iceland	Film	smais.is	L 7 12 14 16 18	all audiences 7 or over 12 or over 14 or over 16 or over 18 or over	0 7 12 14 16 18		
India	Film	cbfcindia.tn.nic.in	U U/A A S	all audiences 12 or over, supervised 18 or over special	0 12PG 18		
Indonesia	Film	lsf.go.id	SU A BO R D	all audiences children parental guidance teen mature	0 13		
Ireland	TV	rte.ie	GA CH YA PS MA	all audiences children young adults parental guidance mature	0		
Ireland	Film	ifco.ie	G PG 12A 15A 16	all audiences parental guidance 12 or over, supervised 15 or over, supervised 16 or over	0 12PG 15PG 16		

Authority / Locale	Media Type	Domain	Valid Ratings	Rating Description	Age Equivalence	Valid Advice	Advice Description
			18	18 or over	18		
Japan	Film	eirin.jp	G	all audiences	0		
			PG-12	12 or over, supervised	12PG		
			R-15	15 or over	15		
			R-18	18 or over	18		
Japan / CERO	Games	cero.gr.jp	A	all audiences	0		
			B	12 or over	12		
			C	15 or over	15		
			D	17 or over	17		
			Z	18 or over	18		
Latvia	Film	nfc.lv	V	all audiences	0		
			VP-10	10 or over, supervised	10PG		
			VP-12	12 or over, supervised	12PG		
			N-12	12 or over	12		
			N-14	14 or over	14		
			N-16	16 or over	16		
			N-18	18 or over	18		
Maldives	Film & TV	nbc.gov.mv	G	all audiences	0		
			PG	parental guidance			
			12+	12 or over	12		
			15+	15 or over	15		
			18+	18 or over	18		
			18+R	18 or over, graphic	18		
			PU	professional use			
Mexico	Film & TV	rtc.gob.mx	AA	children, under 7	0		
			A	all audiences	0		
			B	12 or over	12		
			B-15	15 or over, film only	15		
			C	18 or over	18		
			D	adult			
Netherlands	Film & TV	kijkwijzer.nl	AL	all audiences	0	Violence	
			6	6 or over	6	Scary	
			9	9 or over	9	Sex	
			12	12 or over	12	Discrimination	
			16	16 or over	16	Drugs	
						Language	
New Zealand	Film	censorship.govt.nz	E	exempt			
			G	all audiences	0		
			PG	parental guidance			
			M13	13 or over	0		
			M	mature	0		
			R13	13 or over	13		
			R15	15 or over	15		
			R16	16 or over	16		

Authority / Locale	Media Type	Domain	Valid Ratings	Rating Description	Age Equivalence	Valid Advice	Advice Description
			R18	18 or over	18		
			RP16	16 or over	16		
Nigeria	Film	nfvcb.gov.ng	G	all audiences	0		
			PG	parental guidance			
			12	12 or over	12		
			12A	12 or over, supervised	12PG		
			15	15 or over	15		
			18	18 or over	18		
			RE	restricted			
Norway	Film	medietilsynet.no	A	all audiences	0		
			7	7 or over	7		
			11	11 or over	11		
			15	15 or over	15		
			18	18 or over	18		
Philippines / MTRCB	TV	.ph_MTRCB_TV	General Patronage	all audiences	0		
			Parental Guidance	parental guidance			
Philippines / MTRCB	Film	.ph_MTRCB_FILM	G	all audiences	0		
			GP	all audiences	0		
			PG-13	13 or over, supervised	13PG		
			R	17 or over	17		
			R-13	13 or over	13		
			R-18	18 or over	18		
			X	not for public viewing			
Poland	TV	krrit.gov.pl	Green Circle	all ages			
			Yellow Circle	parental guidance			
			Red Circle	adult			
			Yellow 7	7 or over	7		
			Yellow 12	12 or over	12		
			Yellow 16	16 or over	16		
Poland	Film	.po_ FILM	BO	all audiences	0		
			6	6 or over	6		
			12	12 or over	12		
			15	15 or over	15		
			18	18 or over	18		
			21	21 or over	21		
			Green Circle	all audiences	0		
			Yellow 7	7 or over	7		
			Yellow 12	12 or over	12		
			Yellow 16	16 or over	16		
			Red Circle	18 or over	18		
Portugal	Film	cce.org.pt	M/4	4 or over	4	-Q	quality
			M/6	6 or over	6		

Authority / Locale	Media Type	Domain	Valid Ratings	Rating Description	Age Equivalence	Valid Advice	Advice Description
			M/12	12 or over	12		
			M/16	16 or over	16		
			M/18	18 or over	18		
			M/18-P	18 or over, sexual	18		
Serbia	Film	rra.org.yu	<age>	<age> or over	<all ages valid>		
Singapore	Film	mda.gov.sg	G	all audiences	0		
			PG	parental guidance			
			NC16	16 or over	16		
			M18	18 or over	18		
			R18	18 or over	18		
			R21	21 or over	21		
South Africa	TV	fpb.gov.za_TV	Family	all audiences	0	V	violence
			PG	parental guidance		N	nudity
			13	13 or over	13	S	sex
			15	15 or over	15	L	language
			18	18 or over	18		
			R18	adult			
South Africa	Film	fpb.gov.za_FILM	A	all audiences	0		
			PG	parental guidance			
			10M	10 or over, supervised	10PG		
			10	10 or over	10		
			13	13 or over	13		
			16	16 or over	16		
			R18	18 or over	18		
			X18	18 or over, sexual	18		
Sweden	Film	statensbiografbyra.se	Btl	all audiences	0		
			7 years	7 or over	7		
			11 years	11 or over	11		
			15 years	15 or over	15		
			Prohibited	banned			
Taiwan	Film	gio.gov.tw	General audiences	all audiences	0		
			Protected	6 to 12, supervised	6-12PG		
			Parental guidance	12 to 18, supervised	12-18PG		
			Restricted	18 or over	18		
United Kingdom / British Board of Film Classification	Film & TV	bbfc.co.uk	Uc	all audiences, children	0		
			U	all audiences	0		
			PG	parental guidance			
			12A	12 or over, supervised	12PG		
			12	12 or over	12		
			15	15 or over	15		
			18	18 or over	18		

Authority / Locale	Media Type	Domain	Valid Ratings	Rating Description	Age Equivalence	Valid Advice	Advice Description
			R18	18 or over, sexual	18		
			G	adult			
United Kingdom / ELSPA	Games	elspa.com	3-10	3 or over	3		
			11-14	11 or over	11		
			15-17	15 or over	15		
			18+	18 or over	18		
United States / MPAA	Film	mpaa.org	G	all audiences	0		
			PG	parental guidance			
			PG-13	13 or over, supervised	13PG		
			R	17 or over, supervised	17PG		
			NC-17	17 or over	17		
			NR	not rated			
United States / Film Advisory Board	Film	filmadvisoryboard.org	F	all audiences	0		
			PD	parental guidance			
			PD-M	13 or over	13		
			EM	17 or over	17		
			AO	18 or over	18		
United States / RIAA	Music	riaa.com	PAL	Parental Advisory (explicit content)			
United States / ESRB	Games	esrb.org	EC	children	0		
			E	all audiences	0		
			E10+	10 or over	10		
			T	teen	13		
			M	mature	0		
			AO	adults only			
			RP	rating pending			
Venezuela	TV	leyresorte.gob.ve	A				
			B				
			C				
			D				
			E	sexual			

Annex M
(informative)

Bibliography

Reference documents

The following references, or portions thereof, are cited as required for compliance with this document/standard.

[71] IETF RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, March 1997.
http://www.ietf.org/rfc/rfc2119.txt

[72] IETF RFC 2766, Network Address Translation - Protocol Translation (NAT-PT), G. Tsirtsis, P. Srisuresh, February 2000.
http://www.ietf.org/rfc/rfc2766.txt

[73] IETF RFC 2929, Domain Name System (DNS) IANA Considerations, D. Eastlake, E. Brunner-Williams, B. Manning, September 2000.
http://www.ietf.org/rfc/rfc2929.txt

[74] AHRA, U.S. Audio Home Recording Act of 1992, United States Public Law 102-563, Subchapter D, Section 1008, 1992.
http://thomas.loc.gov/cgi-bin/query/z?c102:S.1623.ENR:

[75] Universal Unique Identifier, DCE 1.1 Appendix for Universal Unique Identifiers, The Open Group, 1997.
http://www.opengroup.org/onlinepubs/9629399/apdxa.htm

[76] 3GPP TS 23.107, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Quality of Service (QoS) concept and architecture (Release 6), December 2004
http://www.3gpp.org/ftp/Specs/archive/23_series/23.107/23107-620.zip

[77] XHTML-Print/CSS Print Guidelines, XHTML-Print/CSS Print Profile Guidelines for PrintEnhanced:1, UPnP Forum, May 4, 2005.
http://www.upnp.org/standardizeddcps/documents/PrintEnhanced1_guideline_v1_050504.pdf

[78] CSS3 Paged Media Module, CSS3 Paged Media Module, H. Lie and J. Bigelow (editor), W3C, February 25, 2004.
http://www.w3.org/TR/2004/CR-css3-page-20040225

[79] BNEP, Bluetooth Network Encapsulation Protocol (BNEP) Specification, version 1.0, Bluetooth SIG, February 14, 2003.
https://www.bluetooth.org/

[80] Bluetooth Security Architecture, Bluetooth Security Architecture, Whitepaper, version 1.0, Bluetooth SIG, July 15, 1999.
https://www.bluetooth.org

[81] PAN Profile, Personal Area Networking Profile version 1.0, Bluetooth SIG, February 14, 2003.
https://www.bluetooth.org

[82] Specification of the Bluetooth System (Volume 1 - Core, Volume 2 - Profiles, Addendum, Errata), Specification of the Bluetooth System, version 1.1, Bluetooth SIG,

February 22, 2001.
https://www.bluetooth.org

[83] Specification of the Bluetooth System, Master Table of Content and Compliance Requirements, version 1.2, Bluetooth SIG, November 05, 2003.
https://www.bluetooth.org

[84] Wi-Fi 802.11with WPA2, WPA, and WEP System Interoperability Test Plan with ASD Test Engine for IEEE 802.11a, b, and gg Devices Version 1,15, Wi-Fi Alliance, January 17, 2011.
http://www.wi-fi.org/testing_information.php

[85] WMM Specification, Wi-Fi WMM (Wireless Multimedia) Specification, Wi-Fi Alliance, March 09, 2004.
http://www.wi-fi.org

[86] WMM Test Plan, Wi-Fi WMM System Interoperability Test Plan, version 2.2, Wi-Fi Alliance, January 12, 2011.
http://www.wi-fi.org/testing_information.php

[87] ANSI/ICEA S-90-661-2002, Category 3, 5, & 5e Individually Unshielded Twisted Pair Indoor Cable for Use In General Purpose and LAN Communication Wiring Systems, Insulated Cable Engineers Association, June 27, 2002.
http://www.icea.net/

[88] UPnP Printer:1, Printer:1 Device Template Version 1.01, Shivaun Albright, Tom Hastings, Harry Lewis, Peter Zehler, UPnP Forum, August 8, 2002.
http://www.upnp.org/standardizeddcps/documents/Printer_Definition_v1_020808.pdf

[89] UPnP Printer:1 Annex A V1.0, Printer:1 Device Template Version 1.01 with Annex A - Optional Service Addition V1.0, UPnP Forum, May 4, 2005.
http://www.upnp.org/standardizeddcps/documents/AnnexA-Printer_v1_050504.pdf

[90] UPnP PrintEnhanced:1, PrintEnhanced:1 Service Template Version 1.01, UPnP Forum, May 4, 2005.
http://www.upnp.org/standardizeddcps/documents/Service_PrintEnhanced_v1_050504.pdf

[91] ASF, Advanced System Format (ASF) Specification, Microsoft Corporation, December 2004.
http://www.microsoft.com/windows/windowsmedia/format/asfspec.aspx.

[92] RTP Payload format for WMV and WMA, RTP Payload Format for Windows Media Audio and Video, Microsoft Corporation.
http://download.microsoft.com/download/5/5/a/55a7b886-b742-4613-8ea8-d8b8b5c27bbc/RTPPayloadFormat_for_WMAandWMV_v1.doc

[93] ATSC Standard A/52B, Digital Audio Compression (AC-3) Rev B, Advanced Television Systems Committee, 14 June. 2005.
http://www.atsc.org/standards/a_52b.pdf

[94] 3GPP TS 26.244, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects;Transparent end to end packet switched streaming service (PSS); 3GPP file format (3GP)(Release 6), September 2005.
http://www.3gpp.org/ftp/Specs/archive/26_series/26.244/26244-640.zip

[95] PWG5101.1,PWG Candidate Standard 5101.1-2002 - Standard for Media Standardized Names, Ron Bergman and Tom Hastings, Printer Working Group, February 26,

2002.
ftp://ftp.pwg.org/pub/pwg/standards/pwg5101.1

[96] XML Schema for ContentDirectory:3 LastChange Event CDS:event-v1-2007 (cds-event-v3-20071128.xsd)
http://www.upnp.org/schemas/specs/av/av3.asp

[97] MoCA MAC/PHY Specification v1.1, Multimedia over Coax Alliance
http://www.mocalliance.org/

[98] MoCA Certification Test Plan v1.1, Multimedia over Coax Alliance
http://www.mocalliance.org/

[99] Wi-Fi Simple Configuration Test Plan, Wi-Fi Alliance, Version 2.2, 24-Feb-20117
http://www.wi-fi.org/testing_information.php

[100] 802.11n System Interoperability Test Plan Version 2,0,10, Wi-Fi Alliance, 09-Feb-2011
http://www.wi-fi.org/testing_information.php

[101] CEA-2014 Revision A, Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740
http://global.ihs.com; E-mail: global@ihs.com

[102] HPNA 3.1 Certification Specification, Version 0.2, October 2006
http://www.homepna.org

[103] Wi-Fi Direct System Interoperability Test Plan, Version 1.0.0, Wi-Fi Alliance, 25-Oct-2010
http://www.wi-fi.org/testing_information.php

[104] ARIB TR B-14 Ver. 2.7, Operational Guidelines For Digital Terrestrial Television Broadcasting, March 14 2006.

[105] ARIB TR B-15 Ver. 3.6, Operational Guidelines For Digital Satellite Broadcasting, March 14 2006.

[106] ANSI/CEA-2033 A, Specification for Electronic Program Guide Data Interchange. ANSI/CEA, March 2008.

[107] ANSI/CEA-766-C, U.S. and Canadian Rating Region Tables (RRT) and Content Advisory Descriptors for Transport of Content Advisory Information Using ATSC Program and System Information Protocol (PSIP), April 2008