DLNA

Digital Living Network Alliance

Home Networked Device Interoperability Guidelines v1.0

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 Home Networked Device Interoperability 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 applications for the digital home.

Do Not Copy

Home Networked Device Interoperability Guidelines v1.0

ABSTRACT

Consumers are acquiring, viewing, and managing an increasing amount of digital media (photos, music, and video) on devices in the Consumer Electronics (CE), mobile, and Personal Computer (PC) domains. As such, they want to conveniently enjoy the content-regardless of the source-across different devices and locations in the home. 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, a common set of industry design guidelines is required that allows vendors to participate in a growing marketplace, leading to more innovation, simplicity, and value for consumers. This document serves that purpose and provides vendors with the information needed to build interoperable networked platforms and devices for the digital home.

LEGAL DISCLAIMER

Use of the information contained herein shall be governed solely by the terms and conditions of the Digital Living Network Alliance IPR Policy. The document and information contained herein is not a license, either expressly or impliedly, to any intellectual property owned or controlled by any of the authors or developers of this specification. The information contained herein is provided on an "AS IS" basis, and to the maximum extent permitted by applicable law, the authors and developers of this specification hereby disclaim all other warranties and conditions, either express, implied or statutory, including but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, of lack of negligence. ALSO THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT.

DLNA is a registered trademark of the Digital Living Network Alliance. *Other names and brands may be claimed as the property of others.

Home Networked Device Interoperability Guidelines v1.0

LIST OF FIGURES

LIST OF TABLES AND CHARTS

1 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-industry 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.

1.1 PURPOSE/SCOPE

This document 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 **Section 4, "DLNA Home Networking Architecture"**) 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, and media transport protocols. **Table 1-1** shows a summary of the key functional components and technology ingredients that are covered by the Interoperability Guidelines.

Table 1-1 Key Technology Ingredients

Functional Components	Technology Ingredients
Connectivity	Ethernet* and 802.11 Media
Networking	IPv4 Suite
Device Discovery and Control	UPnP* Device Architecture v1.0
Media Management and Control	UPnP AV v1.0
Media Formats	Required and Optional Format Profiles
Media Transport	HTTP

1.2 AUDIENCE

This document is 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.

1.3 ORGANIZATION

The remainder of this document is organized as follows:

- **Section 2, "References Acquisition"**: Acquisition information on all normative and informative references contained in this document.
- **Section 3, "Acronyms and Terms"**: Definitions of acronyms and terms used in this document.
- **Section 4, "DLNA Home Networking Architecture"** An overview of the DLNA home networking architecture.
- **Section 5, "Device Classes Overview"**: An overview of the major device categories used to group guideline requirements.
- **Section 6, "Guideline Terminology and Conventions"**: Definitions for the compliance and usage classifications used for guideline requirements.
- **Section 7, "Guideline Requirements"**: Covers guideline requirements for DLNA devices.
- **Appendix 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.
- **Appendix B:** Describes the way DLNA devices should represent tuner-based content.
- **Appendix C:** Describes how a DLNA device can represent itself on multiple network interfaces. The appendix also discusses how a serving endpoint should expose content URI values for different network interfaces.

REFERENCES ACQUISITION

2

2.1 NORMATIVE REFERENCES

The following references, or portions thereof, are cited in the DLNA Home Networked Device Interoperability Guidelines as required for compliance with this document.

[1] 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.

[2] 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.

[3] IEEE 802.11a-1999 (Supplement to IEEE 802.11, 1999 Edition, Reaffirmed 2003),

IEEE Standard for information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHz Band, Reaffirmed June, 12 2003.

[4] IEEE 802.11b-1999 (Supplement to IEEE 802.11, 1999 Edition, Reaffirmed 2003),

IEEE Standard for information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer (PHY) Extension in the 2.4 GHz Band, Reaffirmed June 12, 2003.

[5] IEEE 802.11g-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001),

IEEE Standard for information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access

Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher-Speed Physical Layer (PHY) Extension in the 2.4 GHz Band, June 27, 2003.

[6] XML W3C Recommendations,
Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, October 6, 2000.

[7] Wi-Fi 802.11a and a + b or g WPA Interoperability Test Plan,
Wi-Fi 802.11a with WPA System Interoperability Test Plan for IEEE 802.11a Devices and IEEE 802.11a + b or g Devices Version 2.2, Wi-Fi Alliance, November 28, 2003.

[8] Wi-Fi 802.11b WPA Interoperability Test Plan,
Wi-Fi 802.11b with WPA System Interoperability Test Plan for IEEE 802.11b Devices Version 2.2, Wi-Fi Alliance, November 27, 2003.

[9] Wi-Fi 802.11g WPA Interoperability Test Plan ,
Wi-Fi 802.11g with WPA System Interoperability Test Plan Version 2.2, Wi-Fi Alliance, November 20, 2003.

[10] ASD Test Plan ,
WECA Interoperability Test Plan Application Specific Devices Version 2.0, Wi-Fi Alliance, December 2, 2001.

[11] IETF RFC 768,
User Datagram Protocol, J. Postel, August 28, 1980.

[12] IETF RFC 791,
Internet Protocol, J. Postel, September 1981.

[13] IETF RFC 792,
Internet Control Message Protocol, J. Postel, September 1981.

[14] IETF RFC 793,
Transmission Control Protocol, J. Postel, September 1981.

[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.

[16] IETF RFC 1122,
Requirements for Internet Hosts – Communications Layers, R. Braden, October 1989.

[17] IETF RFC 1812,
Requirements for IP version 4 Routers, F. Baker, June 1995.

[18] IETF RFC 1738,
Uniform Resource Locators (URL), T. Berners-Lee, CERN, L. Masinter, Xerox Corporation, M. McCahill, University of Minnesota, December 1994.

[19] IETF RFC 1945,
Hypertext Transfer Protocol – HTTP/1.0, T. Berners-Lee, MIT/LCS, R. Fielding, UC Irvine, H. Frystyk, May 1996.

[20] IETF RFC 2131,
Dynamic Host Configuration Protocol, R. Droms, March 1997.

[21] 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.

[22] IETF RFC 2279,
UTF-8, a transformation format of ISO* 10646, F. Yergeau, Alis Technologies, January 1998.

[23] 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.

[24] 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.

[25] IETF RFC 2822,
Internet Message Format, P. Resnick, QUALCOMM Incorporated, April 2001.

[26] Auto IP,
Dynamic Configuration of IPv4 Link-Local addresses, Stuart Cheshire, November 24, 2000, expires May 24, 2001.

[27] ISO 8601:2000,
Data elements and interchange formats – Information interchange -- Representation of dates and times, International Standards Organization, December 21, 2000.

[28] UPnP Device Architecture Version 1.0,
UPnP Device Architecture Version 1.0, UPnP Forum, June 13, 2000.

[29] UPnP AVTransport:1,
AVTransport:1 Service Template Version 1.01, UPnP Forum, June 25, 2002.

[30] UPnP ConnectionManager:1,
ConnectionManager:1 Service Template Version 1.01, UPnP Forum, June 25, 2002.

[31] UPnP ContentDirectory:1,
ContentDirectory:1 Service Template Version 1.01, UPnP Forum, June 25, 2002.

[32] UPnP MediaRenderer:1,
MediaRenderer:1 Device Template Version 1.01, UPnP Forum, June 25, 2002.

[33] UPnP MediaServer:1,
MediaServer:1 Device Template Version 1.01, UPnP Forum, June 25, 2002.

[34] UPnP RenderingControl:1,
RenderingControl:1 Service Template Version 1.01, UPnP Forum, June 25, 2002.

[35] IETF RFC 3555,
MIME Type Registration of RTP Payload Formats, S. Casner, Packet Design, P. Hoschka, July 2003.

[36] ISO/IEC 10918-1:1994,
Information technology - Digital compression and coding of continuous-tone still images: Requirements and guidelines, International Standards Organization.

[37] JEITA CP-3451,
Exchangeable image file format for digital still cameras: Exif Version 2.2, Standard of Japan Electronics and Information Technology Industries Association, April 2002.

[38] *JEITA CP-3451-1*,
Exchangeable image file format for digital still cameras: Exif Version 2.21. (Amendment Ver2.2), Standard of Japan Electronics and Information Technology Industries Association, September 2003.

[39] JPEG File Interchange Format,
Version 1.02, Eric Hamilton, C-Cube Microsystems, September 1, 1992.

[40] JEITA CP-3461,
Design rule for Camera File system DCF Version 2.0, Standard of Japan Electronics and Information Technology Industries Association, September 2003.

[41] ISO/IEC 13818-1:2000,
Information technology -- Generic coding of moving pictures and associated audio information: Systems, International Standards Organization.

[42] ISO/IEC 13818-2:2000,
Information technology -- Generic coding of moving pictures and associated audio information: Video, International Standards Organization.

[43] ISO/IEC 13818-3:1998,
Information technology -- Generic coding of moving pictures and associated audio information: Audio, International Standards Organization.

[44] ISO/IEC 11172-3:1993,
Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s -- Part 3: Audio (MPEG-1Audio), International Standards Organization.

[45] ATSC* Standard A/52A[1],
Digital Audio Compression (AC-3*) Rev A, Advanced Television Systems Committee, August 20, 2001.

[46] ISO/IEC 13818-11:2004,
Information technology -- Generic coding of moving pictures and associated audio information – Part 11:IPMP on MPEG-2 Systems:, International Standards Organization.

[47] DVD* Specifications for DVD-RAM/DVD-RW/DVD-R for General Discs Part 3 Video Recording,
DVD Forum, Version 1.1, May 2001.

1. The ATSC and ITU references are substantially the same and are intended to be identical. From the perspective of interoperability, it doesn't matter which reference is used. The ATSC document has a couple of additional appendices (one is informative and one with additional, optional metadata). Some references in these guidelines reference the ATSC document while other reference the ITU.

[48] IETF RFC 3551,
RTP Profile for Audio and Video Conferences with Minimal Control, H. Schulzrinne and S. Casner, July 2003.

[49] ISO/IEC 15948:2004 (E),
Information technology – Computer graphics and image processing – Portable Network Graphics (PNG): Functional specification, International Standards Organization.

[50] W3C PNG Recommendations,
Portable Network Graphics (PNG) Specification (Second Edition) Information technology – Computer graphics and image processing – Portable Network Graphics (PNG): Functional specification, ISO/IEC 15948:2003 (E), W3C, November 10, 2003.

[51] ATSC Standard A/53C.
Digital Television Standard, Revision C, Advanced Television Systems Committee, May 21, 2004.

[52] CEA*-708-B,
Digital Television (DTV) Closed Captioning, Consumer Electronics Association, December 29, 1999.

[53] ITU-R Recommendation BS.1196-1[1],
Audio coding for digital terrestrial television broadcasting, Annex 2 (Dolby* AC-3 Audio), International Telecommunication Union, April 2001.

[54] ETSI TR 101 154 V1.4.1 (2000-07),
Digital Video Broadcasting (DVB*);Implementation Guidelines for the use of MPEG-2 Systems, Video and Audio in satellite, cable and terrestrial broadcasting applications (DVB Implementation Guidelines), European Telecommunications Standard Institute.

[55] ETSI EN 300 468 V1.5.1 (2003-05),
Digital Video Broadcasting (DVB); Specification for Service Information (SI) In DVB systems (SI), European Telecommunications Standard Institute.

[56] ETSI EN 300 743 V1.2.1 (2002-10),
Digital Video Broadcasting (DVB); Subtitling systems (DVB Subtitles), European Telecommunications Standard Institute.

[57] ETSI EN 301 775 V1.2.1 (2003-05),
Digital Video Broadcasting (DVB); Specification for the carriage of Vertical Blanking Information (VBI) data in DVB bitstreams (DVB VBI data), European Telecommunications Standard Institute.

[58] ETSI EN 300 472 V1.3.1 (2003-05),
Digital Video Broadcasting (DVB); Specification for conveying ITU-R System B Teletext in DVB bitstreams (DVB Teletext), European Telecommunications Standard Institute.

[59] ETSI EN 301 192 V1.3.1 (2003-05),
DVB Specification for Data broadcasting (DVB data broadcasting), European Telecommunications Standard Institute.

[60] ETSI ES 201 812 V1.1.1 (2003-12),
Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) specification 1.0.3 (DVB-MHP), European Telecommunications Standard Institute.

[61] CEA-849A,
Application Profiles for EIA*-775A compliant DTVs, CEA. (DTV), Consumer Electronics Association, December 6, 2001.

[62] SCTE 54 2002 (formerly DVS 241),
Digital Video Service Multiplex and Transport System Standard for Cable Television, Society of Cable Telecommunications Engineers Inc.

[63] ID3 tag version 2.4.0 – Main Structure,
M. Nilsson, November 1, 2000.

[64] IETF RFC 3003,
The audio/mpeg Media Type, M. Nilsson, November 2000.

[65] ARIB* STD-B24 Version 3.2,
Data Coding and Transmission Specification for Digital Broadcasting, Association of Radio Industries and Businesses, November 15, 2001.

2.2 INFORMATIVE REFERENCES

The following documents contain information that is useful in understanding the DLNA Home Networked Device Interoperability Guidelines.

[66] DH White Paper,
Digital Home White Paper, Digital Living Network Alliance, June 2003.

[67] IEEE 802.1D-1998, Annex H, *I*
EEE 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 25, 1998.

[68] IETF RFC 2119,
Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, March 1997.

[69] IETF RFC 2766,
Network Address Translation - Protocol Translation (NAT-PT), G. Tsirtsis, P. Srisuresh, February 2000.

[70] IETF RFC 2929,
Domain Name System (DNS) IANA Considerations, D. Eastlake, E. Brunner-Williams, B. Manning, September 2000.

[71] AHRA,
U.S. Audio Home Recording Act of 1992, United States Public Law 102-563, Subchapter D, Section 1008, 1992.

[72] Universal Unique Identifier,
DCE 1.1 Appendix for Universal Unique Identifiers, The Open Group, 1997.

[73] ETSI TR 101 202 V1.2.1 (2003-01),
Digital Video Broadcasting (DVB); Implementation guidelines for Data Broadcasting, European Telecommunications Standard Institute, January 2003.

3 ACRONYMS AND TERMS

3.1 ACRONYMS

The following acronyms are used in the DLNA Home Networked Device Interoperability Guidelines.

Table 3-1 DLNA Acronym Usage

Acronym	Definition
AC-3	Popularly known as Dolby Digital*, an audio format standard for delivering up to 5.1 audio channels developed by Dolby Laboratories.
ACK	"Acknowledge" Typically used to describe an action following a network packet being successfully received.
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.
ARIB	"Association of Radio Industries and Businesses" One of the standard bodies for digital television broadcasting.
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.
ATSC	"Advanced Television Systems Committee" One of the standard bodies for digital television broadcasting.
AV	"Audio with Video" Refers to any media content that contains both moving picture and sound.
AVT	"AVTransport Service 1.0" 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.
CAT	"Conditional Access Table"
CBR	"Constant Bit Rate"
CDS	"ContentDirectory Service 1.0" The ContentDirectory Service is a UPnP service that provides network-based discovery of content. The ContentDirectory Service specification is a standard UPnP DCP.
CE	"Consumer Electronics" A class of devices used in the home, such as DVD, DVR, PVR, PDA, TV, set-top box, cellular phones, …

Table 3-1 DLNA Acronym Usage (Continued)

Acronym	Definition
CMS	"ConnectionManager Service 1.0" 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.
CP	"UPnP Control Point" Generic reference to any UPnP control point.
DA	"Device Architecture 1.0 UPnP Device Architecture version 1.0 document.
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 1.0 DCP.
DDC	"Device Discovery and Control" A section heading in the Interoperability Guidelines.
DHCP	"Dynamic Host Configuration Protocol" A protocol to automatically provide IP addresses and other network configuration information to network nodes.
DMP	"Digital Media Player" DLNA device class implementation for an MSCP with unexposed MRD functionality.
DMS	"Digital Media Server" DLNA device class implementation for an MSD.
DLNA	"Digital Living Network Alliance" The organization that created this document.
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.
DVB	"Digital Video Broadcast" One of the standard bodies for digital television broadcasting.
DVD	"Digital Versatile Disc" A high capacity multimedia data storage medium.
DVR	"Digital Video Recorder" A consumer electronic device.
EDTV	"Enhanced Definition Television"
EXIF	"Exchangable Image File"
ES	"Elementary Stream" A single stream of several MPEG-2 carried components.
HD	"High Definition" Picture quality at an HDTV level.
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.

Table 3-1 DLNA Acronym Usage (Continued)

Acronym	Definition
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.
ICMP	"Internet Control Message Protocol" A protocol in the TCP/IP family that is used for out-of-band messages related to network operation.
IGD	"Internet Gateway Device" A multifunction network infrastructure device that routes and/or bridges global internet with the local area network.
IPR	"Intellectual Property Rights"
IP	"Internet Protocol"
IPv4	"Internet Protocol version 4" An OSI network layer 3 protocol.
JPEG	"Joint Photographic Experts Group" It is a coding standard for compression of still images (pictures).
LAN	"Local Area Network" Closely administered network segment(s) such as within the home or office.
LFE	"Low Frequency Enhancement" DVB-specified way to transmit additional sound information.
LPCM	"Linear Pulse Code Modulation" An uncompressed audio encoding.
MF	"Media Formats" A section heading in the Interoperability Guidelines.
MHP	"Multimedia Home Platform" An optional application interface used together with MPEG-2 transmissions.
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.
MM	"Media Management" A section heading in the Interoperability Guidelines.
MPEG-1	"Moving Picture Experts Group phase 1"
MPEG-2	"Moving Picture Experts Group phase 2"
MRCP	"MediaRenderer 1.0 Control Point" UPnP control point that issues actions to an MRD.
MRD	"MediaRenderer Device 1.0" 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.
MSCP	"MediaServer 1.0 Control Point" UPnP AV control point that issues actions to an MSD.

Table 3-1 DLNA Acronym Usage (Continued)

Acronym	Definition
MSD	"MediaServer Device 1.0" 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.
MT	"Media Transport" A section heading in the Interoperability Guidelines.
NC	"Networking and Connectivity" A section heading in the Interoperability Guidelines.
NIC	"Network Interface Controller (or Card)" Hardware that provides an interface for a device to a network such as an Ethernet network.
NTSC*	"National Television Systems Committee" A standard for broadcast and reception of analog television signals.
OSI	"Open Systems Interconnection" Networking stack model (7 layers).
PAL*	"Phase Alternating Line" A standard for broadcast and reception of analog television signals.
PAT	"Program Association Table" A table that lists all the services (programs) in a Transport Stream.
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.
PMT	"Program Map Table" A table that identifies all the elementary streams that belong to the same service (program) in a Transport Stream.
PNG	"Portable Network Graphics" It is a coding standard for compression of still images (pictures).
PS	"Program Stream" Usually in reference to an MPEG-2 AV stream format.
PVR	"Personal Video Recorder" A consumer electronic device.
QoS	"Quality of Service" To provide guarantees on the ability of a network to deliver predictable results.
RCS	"RenderingControl Service 1.0" 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.
SCPD	"Service Control Protocol Description" This is an XML-encoded file describing a UPnP service. This is also known as a service description file.
SD	"Standard Definition" Picture quality at an SDTV level.

Table 3-1 DLNA Acronym Usage (Continued)

Acronym	Definition
SDTV	"Standard Definition Television" Mode of operation of digital television that provides standard quality display, with a vertical resolution display less than 720p and an aspect ration of 4:3, resulting in a viewing experience similar or slightly better than today's analog television.
SIT	"Selection Information Table" SIT describes the service(s) and event(s) carried by a partial TS.
SOAP	"Simple Object Access Protocol" An XML-based messaging protocol used to exchange service requests and responses over a network.
SPTS	"Single Program Transport Stream"
SSDP	"Simple Service Discovery Protocol" UPnP device discovery protocol.
TCP	"Transmission Control Protocol" A protocol in the TCP/IP family used for the reliable exchange of data over a network.
TIFF	"Tagged Image File Format" Media format of still images (pictures).
TS	"Transport Stream" Usually in reference to an MPEG-2 AV stream format.
TTS	"Timestamped Transport Stream"
UDP	"User Datagram Protocol" A protocol in the TCP/IP family used for the unreliable exchange of data over a network.
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.
VBI	"Vertical Blanking Interval"
VBR	"Variable Bit Rate"
VOB	"Video Object"
WAN	"Wide Area Network" Usually in reference to the network outside the home or office. Typically in reference to the entire global Internet.
XML	"Extensible Markup Language" A text-based declarative language used to describe structured data for information exchange.

3.2 DEFINITION OF TERMS

The following terms are used in the DLNA Home Networked Device Interoperability Guidelines.

Table 3-2 DLNA Term Usage

Term	Definition
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”.
Device Class	Defines specific functionality supported on a device regardless of its physical attributes. Examples used within this document are DMS (Digital Media Server) and DMP (Digital Media Player). Each of these device classes defines a specific collection of functions supported by the device. A single device that supports multiple functions may fall into multiple Device Classes. See **Section 5, “Device Classes Overview”** for more information.
Device Type	A specific category of device defined by its functionality, its physical attributes, and its usage models. DVD players, TVs, DVRs, Mobile Phones, Printers, Cameras, Picture Frames, etc. are all examples.
Exposed	Content that is listed by a UPnP AV ContentDirectory Service (CDS). Content may not necessarily exist at the time of “exposition” (i.e. will require transcoding or conversion).
ID3, ID3v2	A general tagging format for audio that makes it possible to store meta data about the audio inside the audio file itself. It is a tag mainly targeted at files encoded with MPEG-1/2 layer I, MPEG-1/2 layer II, MPEG-1/2 layer III, and MPEG-2.5, but may work with other types of encoded audio or as a stand alone format for audio meta data.
kbps	Refers to the number of kilobits per second (1 kbps = 1,000 bits per second).
kHz	Refers to the number of kilohertz (1 kHz = 1,000 hertz).
Mbps	Refers to the number of megabits per second (1 Mbps = 1,000,000 bits per second).
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).
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.
Rendering Endpoint	A term used to denote any device class that receives content for playback/streaming scenarios. For example, the DMP device class.
Serving Endpoint	A term used to denote any device class that serves content for playback/streaming scenarios. For example, the DMS device class that serves content to the DMP device class.

Table 3-2 DLNA Term Usage

Term	Definition
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.

DLNA Home Networking Architecture

4

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 1-1 in Section 1 shows the specific functional components and technology ingredients that are covered in the Interoperability Guidelines. Figure 4.1. 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 Section 5. A brief overview of each functional component follows in the subsequent subsections.

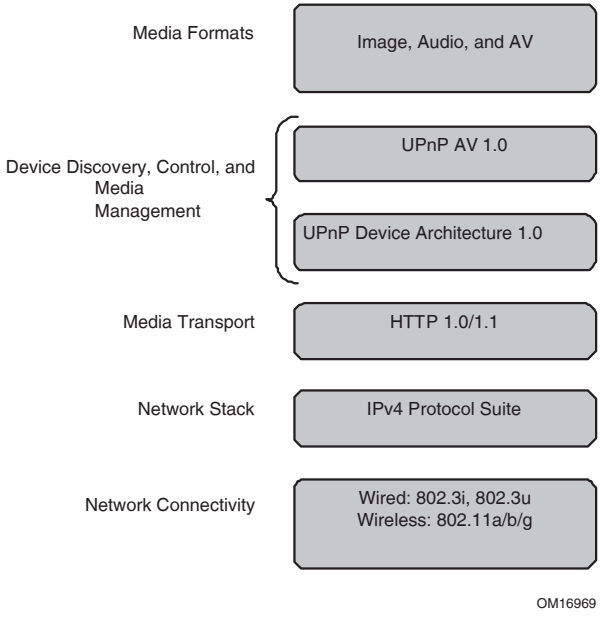


Figure 4.1. DLNA Functional Components

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 are Ethernet (802.3i and 802.3u) and wireless technologies (802.11a, 802.11b, and 802.11g).

Section 7.1 specifies the detailed guidelines to enable interoperability between DLNA devices in the digital home. In addition, the home network requires supporting network infrastructure devices, such as Internet gateways, routers, and switches. Appendix A provides recommendations for network infrastructure devices to facilitate a good user experience with DLNA devices.

4.2 DEVICE DISCOVERY AND CONTROL

Device discovery and control enables a device on the home network to automatically self-configure networking properties such as an IP address, 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. Section 7.2 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) 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. Examples of UPnP AV devices include TVs, VCRs, DVD players, Set-Top Boxes, stereo systems, still-image cameras, 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: Enumerates 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.**

This version of the DLNA Home Networked Device Interoperability Guidelines only supports interaction scenarios between UPnP MediaServer devices and control points. Network-based control of rendering provided by UPnP MediaRenderer devices and control points is not in scope. Future versions may have guidelines that support interaction scenarios involving UPnP MediaRenderers. See Section 5, "Device Classes Overview" for further information on how UPnP technology components are mapped into DLNA device classes.

Section 7.3 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. 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 item to enable interoperability between DLNA devices in the digital home. 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. Section 7.4 through Section 7.7 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 render media content across the home network must support HTTP as the baseline transport mechanism for media streaming or transfer. Section 7.7 specifies the detailed guidelines to enable interoperability between DLNA devices in the digital home.

DEVICE CLASSES OVERVIEW

5

As described in Section 4, "DLNA Home Networking Architecture", devices adhering to the DLNA Home Networked Device Interoperability Guidelines have two major technology components that rely on existing standards:

- UPnP AV and the UPnP Device Architecture for device discovery, control, and media management.
- HTTP for media transport.

The UPnP AV Architecture defines two types of AV devices:

- UPnP AV MediaServer Device (MSD), a device that sources media.
- UPnP AV MediaRenderer Device (MRD), a device that is a sink for the media.

An MSD and an MRD also have corresponding control points: a MediaServer Control Point (MSCP) and a MediaRenderer Control Point (MRCP). Although the UPnP AV Architecture allows for very general interaction scenarios that involve an MSD, MRD, and a combined MSCP+MRCP, this version of the Interoperability Guidelines only supports restricted interaction scenarios between an MSD and an MSCP. Future versions of the Interoperability Guidelines may support other interaction scenarios allowed by the UPnP AV Architecture.

A device class is a fundamental concept for the Interoperability Guidelines and is used to define specific functionality supported on a device regardless of its physical attributes. For example, a device class may be defined for the functionality of serving content. A device class's functionality maps into needed technology components such as those described above. For example, the functionality of serving content requires a UPnP AV MediaServer device, an HTTP server for media transport, and so on. Guidelines then place requirements on technology components that are applicable to device classes that need them. Note that the device classes do not differentiate which media classes (see Section 3, "Acronyms and Terms"), such as audio only, are supported by a device. In this version of the Interoperability Guidelines the following device class functionalities are supported:

- Serve media (MSD + HTTP Server); and
- Select, control and render the selected media (MSCP + HTTP Client).

Figure 5.1. defines the device classes used in this document. Essentially, the DMS fulfills the role of distributing content throughout the home. The DMP fulfills the role of finding content exposed by a DMS and playing that content locally on the DMP.

Table 5-1 DLNA Device Classes

DLNA Device Class	UPnP AV Components	Media Transport Components	Functional Description
DMS (Digital Media Server)	MSD	HTTP Server	Serves up media
DMP (Digital Media Player)	MSCP	HTTP Client	Selects, control and render the selected media

Figure 5.1. illustrates the device class components and the device class interaction model supported by the Interoperability Guidelines.

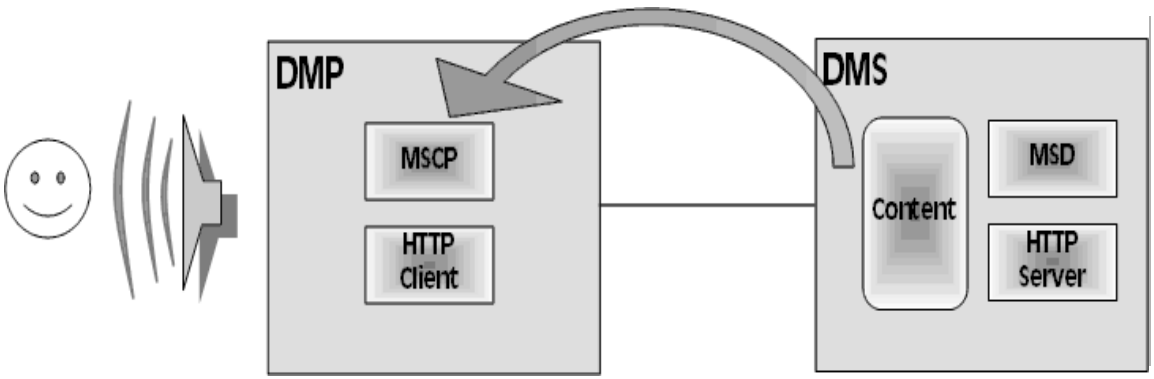


Figure 5.1. DLNA Device Class Components and Interaction Model

GUIDELINE TERMINOLOGY AND CONVENTIONS

6

6.1 GUIDELINE COMPLIANCE CLASSIFIERS

Reference [68] 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]ust, 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 must be implemented", means that the requirement "Y" must 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 must comply with the requirement to ensure interoperability with other implementations.
- **E[X]pressly Forbidden:** This term means that an item must not be incorporated in a product implementation.

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 may be guideline requirements 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 requirement to a source standard or specification:

- **[A]dding:** A guideline requirement that adds to or supplements a standard or specification to enhance interoperability.
- **[C]larifying:** A guideline requirement that addresses vague or ambiguous aspects of a standard or specification.
- **[F]ixing:** A guideline requirement 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 requirement 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 requirement that repeats what is already in a standard or specification because of observed and repeated problems with implementations. Whenever a guideline requirement with this usage classifier seems to be in conflict with the actual standard, the standard prevails over the guideline requirement.

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]**, …
- HTTP headers and methods are always in **bold** font, such as **CACHE-CONTROL**.
- UPnP action names are indicated as: [Service acronym]:[action name], such as CDS:Browse.
- Special terms may be *italicized*. Sometimes a guideline requirement 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.

GUIDELINE REQUIREMENTS

7

This section 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.

Guidelines shown in tabular form in the following sections have a heading layout and column definitions as shown in Figure 7.1.

Figure 7.1. Guideline Table Header Layout and Definitions

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments

- **Name**: A unique label for the guideline. The label is preceded with a sequentially increasing number to allow easy lookup.
- **M/S/O**: The compliance classifier for the guideline. See section Section 6, "Guideline Terminology and Conventions" for the definition of guideline compliance classifiers.
- **A/C/F/L/R**: The specification usage classifier for the guideline. See section 6.2 for the definition of specification usage classifiers.
- **Dev Class**: The device classes that the guideline applies to. See 5-1 for the definition of the DLNA device classes. In addition, the terms "All" and "N/A" are used when a guideline applies to all defined DLNA device classes and when a guideline is definitional in nature (i.e. it will be referenced by subsequent guidelines that apply to specific DLNA device classes), respectively.
- **Description**: The actual description of a guideline. A guideline is preceded with a sequentially increasing number to allow easy lookup. A given guideline may consist of several sub-requirements that are also numbered.
- **Ref #**: Standards that are referenced by the guideline. Standards citations are by number and are defined in section 2.
- **Comment**: Supplementary information about a guideline such as a justification for the guideline, the specific interoperability issue that is addressed, etc.

7.1 NETWORKING AND CONNECTIVITY

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 may 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 varieties of connectivity options for DLNA devices.

The guidelines in Table 7-1 are divided into two sections. The first section provides requirements for general capabilities. For example, these requirements describe the baseline capabilities of any Ethernet implementation. The second section describes the capabilities or combination of capabilities that DLNA devices specifically support. These specific device requirements reference the general capabilities defined in the first section. For example, a specific requirement for DLNA devices is that they must support either Ethernet or 802.11 connectivity, where Ethernet and 802.11 capabilities are described in the first section and are referenced by name.

Table 7-1 Networking and Connectivity Guidelines

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
General Capability Requirements						
7.1.1 NC Ethernet: Base	M	R	N/A	7.1.1.1 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.	[1]	
7.1.2 NC Ethernet: Cabling	S	R	N/A	7.1.2.1 If Ethernet is supported, any supplied network cabling should have a rating of Category 5e or better.	[2]	
7.1.3 NC Ethernet: Gigabit	S	R	N/A	7.1.3.1 If Ethernet is supported, IEEE 802.3ab (1000BASE-T) is recommended in addition to **7.1.1**. An implementation must support auto-negotiation of gigabit operation with a similarly capable link partner and drop down to a lower speed as appropriate.	[1]	Gigabit Ethernet is becoming available and affordable for home networks.
7.1.4 NC Ethernet: QoS Tolerance	M	R	N/A	7.1.4.1 If Ethernet is supported, incoming tagged packets must be tolerated. Tagged packets are Ethernet packets that include priority tags conformant with **[1]**, section 3.5, entitled 'Elements of the Tagged MAC Frame'. Here, 'tolerate' means that the packet payload of any received tagged or untagged packet must be properly passed up to the higher layers in the network stack.	[1]	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 must be able to tolerate them. As home networks are asked to carry more high-bandwidth media streams, QoS handling will become increasingly useful and may be included in the Interoperability Guidelines at a later date.

Table 7-1 Networking and Connectivity Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.1.5 NC 802.11: Base	M	R	N/A	7.1.5.1 If 802.11 is supported, one or more of the following radio selections is allowed: • 802.11a • 802.11b • 802.11g For example, 802.11a, 802.11b, 802.11g, 802.11a/b, 802.11b/g, and 802.11a/b/g all meet this requirement. 7.1.5.2 If 802.11 is supported, the implementation must support infrastructure mode operation.	[3] [4] [5]	There is no implied requirement that the device neither needs to support multiple radios nor is it prohibited.See **Appendix A** for recommendations on Wireless Access Points and how they will help enable interoperability between products with different radio selections. Some DLNA device classes may 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 should assume infrastructure mode as the default.
7.1.6 NC 802.11: Wi-Fi Conformance	M	R	N/A	7.1.6.1 If 802.11 is supported, the implementation must conform to Wi-Fi test plan requirements at the time the product is offered to the market.	[7] [8] [9] [10]	Wi-Fi 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. Examples of these features which are not tested by Wi-Fi at this writing, but which are expected to be in the near future, include advanced link security and link-level QoS. When these capabilities are added to the Wi-Fi certification test plans, wireless implementations must conform to them.

Table 7-1 Networking and Connectivity Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
Device Requirements						
7.1.7 NC Devices: Required Connectivity	M	R	DMS, DMP	7.1.7.1 DLNA device classes must support at least one of the following connectivity selections: Ethernet conformant to all [NC Ethernet:] labeled requirements in the General Capability Requirements section of this table. 802.11 conformant to all [NC 802:11:] labeled requirements in the General Capability Requirements section of this table.		
7.1.8 NC Devices: Recommended Connectivity	S	R	DMS, DMP	7.1.8.1 DLNA device classes should support all of the following connectivity selections: Ethernet conformant to all [NC Ethernet:] labeled requirements in the General Capability Requirements section of this table. 802.11 conformant to all [NC 802:11:] labeled requirements in the General Capability Requirements section of this table. Any of the above selections can be supported via an add-on card, dongle, or equivalent.		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.
7.1.9 NC Devices: IP Stack	M	R	DMS, DMP	7.1.9.1 DLNA device classes must support a TCP/IP stack that includes IPv4, TCP, UDP, ARP, and ICMP components conformant to all required client aspects of **[16]**	**[11]** **[12]** **[13]** **[14]** **[15]** **[16]**	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.
7.1.10 NC Devices: IP Address Acquisition	M	R	DMS, DMP	7.1.10.1 DLNA device classes must support DHCP client functionality **[20]** and obtain an IP address and subnet mask from a home network DHCP server if present. They must implement Auto-IP as defined by the UPnP Device Architecture v1.0 specification (**[26]**,**[28]**) so that if a DHCP server is not present on the home network, a private network address may be automatically acquired.	**[20]** **[26]** **[28]**	

7.2 DEVICE DISCOVERY AND CONTROL

This section 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 section apply only to UPnP-related transactions and not to content playback/streaming transactions. The relevant HTTP-based UPnP transactions are limited to the TCP-based HTTP transactions, such as UPnP XML files (device and service descriptions), SOAP actions, and GENA events. UPnP-related transactions do not include SSDP messages, which are UDP-based.

In this section, 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 section do not apply to HTTP transport for content transfers or playback

.

Table 7-2 Device Discovery and Control Guidelines

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.2.1 DDC UPnP Device Architecture	M	R	DMS, DMP	7.2.1.1 DLNA device classes must fully support the applicable mandatory portions of the UPnP Device Architecture v1.0 (UPnP DA) for discovery, description, control, eventing, and presentation.	[28]	DLNA specifies UPnP Device Architecture 1.0 (UPnP DA) as the basic protocol framework for device classes.
	M	C	DMP	7.2.1.2 A UPnP control point designed for a version of a UPnP Device Architecture, must 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 must be able to do those same things with a UPnP device with a newer minor revision of the UPnP Device Architecture.	[28]	Section 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 7.2.1.2 and 7.2.1.3 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 7.2.1.2 is for the UPnP control point to treat the minor version a UPnP device architecture as 0 (i.e. ignore the minor version).

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	C	DMP	7.2.1.3 A UPnP control point designed for a version of a UPnP device type or service type must 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 must be able to discover and invoke CDS:Browse on a newer UPnP MediaServer. The newer UPnP device may be newer because the device type and/or one of its associated UPnP services are of a newer version.		Section 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. One way to implement 7.2.1.3 is for the UPnP control point to treat any version for the UPnP device and service types as a "1" (i.e. ignore the version number).
	O	C	DMP	7.2.1.4 If an 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 service version.		In other words, if an SOAP action was defined in the specification of a previous service version (e.g. version 1), then a UPnP control point may invoke the SOAP action with the earlier service version regardless of the service version described in the device description.
7.2.2 DDC UPnP Auto IP Support	M	R	DMS, DMP	7.2.2.1 UPnP devices and control points, except those that are acting as a DHCP server, must fully support the Auto-IP specification. See http://www.upnp.org/download/draft-ietf-zeroconf-ipv4-linklocal-01-Apr.txt for further information on Auto IP.	[28]	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.

Table 7-2 Device Discovery and Control Guidelines (Continued)

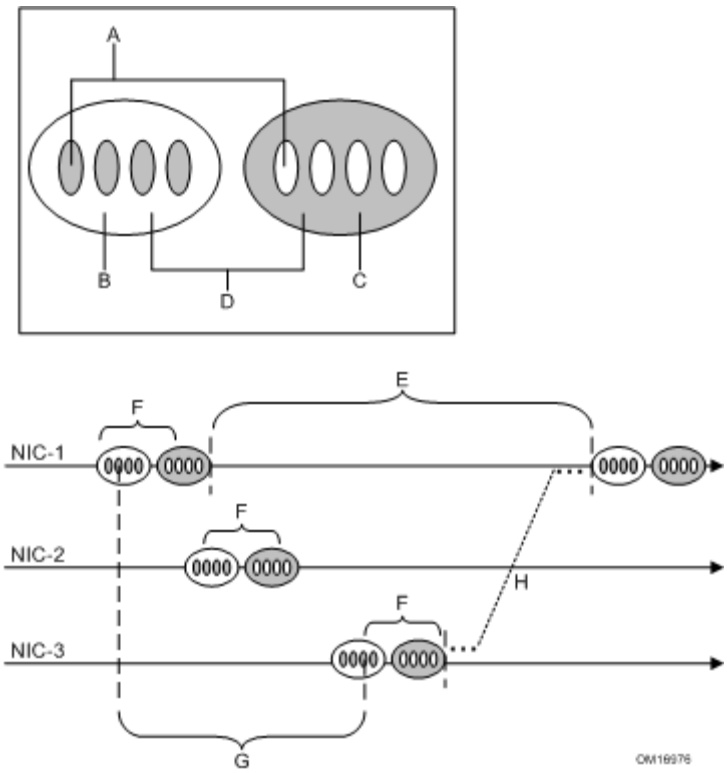
Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	S	L	DMS	7.2.2.2 Whenever a UPnP device switches to a new IP address (whether assigned through Auto-IP or DHCP), the device should send an ssdp:byebye message for (and on) the old IP address. *For (and on) the old IP address* means that the IP address indicated in the (UDP header of the) ssdp:byebye matches the old IP address of the UPnP device.		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.
	M	R	DMS	7.2.2.3 If a UPnP device uses a self-assigned IP address, then the UPnP device must implement duplicate address detection before assigning itself the address.		This guideline repeats a UPnP DA requirement that prevents the assignment of conflicting IP addresses.
7.2.3 DDC UPnP SSDP Default Port	M	C	DMS	7.2.3.1 UPnP devices must receive and process multicast discovery messages on port 1900.	[28]	These requirements ensure that UPnP endpoints will be more likely to discover each other on the home network.
	M	R	DMS	7.2.3.2 UPnP devices must always explicitly specify port 1900 in every **HOST** header tag for every SSDP message.		
	M	R	DMP	7.2.3.3 UPnP control points receiving an SSDP message without the port number in the **HOST** header tag, must assume port number 1900.		
	M	L	DMP	7.2.3.4 UPnP control points must send M-SEARCH messages using a source port greater than 1024 and not 1900.		These guidelines are based on a Microsoft technical advisory regarding security concerns for UPnP.
	O	L	DMS	7.2.3.5 UPnP devices may ignore M-SEARCH messages if the originating source port is less than or equal to 1024 or equal to 1900.		
7.2.4 DDC UPnP Discovery Robustness	S	L	DMS, DMP	7.2.4.1 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.	[28]	This suggestion avoids SSDP discovery flooding on home networks that contain a large number of UPnP endpoints.

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	L	DMS	7.2.4.2 UPnP network devices must not send more than 10 ssdp:alive messages on a single network interface in any given 200 ms period.	[28]	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.
	M	C	DMS	7.2.4.3 UPnP devices must 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 sets* are referred to as *duplicate sets*. The transmission windows for *advertisement sets* and *duplicate sets* cannot overlap in time.	[28]	This guideline clarifies how a UPnP device needs to retransmit its advertisements. However, implementers should remember that advertising too frequently runs the risk of flooding the SSDP channel.
	M	C	DMS	7.2.4.4 A UPnP device that uses the same UDN on multiple network interfaces, must 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.	[28]	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.
	M	C	DMS	7.2.4.5 The interval of sending these *advertisement groups* on a single network interface must 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*.	[28]	For consistency and interoperability, devices need to advertise more often than their notification cycle. However, implementers should remember that advertising too frequently runs the risk of flooding the SSDP channel.
	S	R	DMS	7.2.4.6 The **CACHE-CONTROL** value should be at least 1800, as recommended in the UPnP device architecture.	[28]	Most devices that remain on the network for long periods should have **CACHE-CONTROL** value of 1800. However, some devices (mobile, wireless, etc.) that may want a smaller **CACHE-CONTROL** value.

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments



H — One or more ssdp:alive messages, within *advertisement sets* and *duplicate sets*.
I — *Advertisement set* of 3+2d+k ssdp:alive messages
J — *Duplicate set* of 3+2d+k ssdp:alive messages. (see 7.2.4.3)
K — Combined *advertisement set* and *duplicate sets* make an *advertisement group*. (see 7.2.4.5)
L — Delay between *advertisement groups* on same network is less than ½ of CACHE-CONTROL value. (see 7.2.4.5)
M — Any arbitrary window of 200 ms have 10 or fewer ssdp:alive messages. (see 7.2.4.2) An entire *advertisement set* need not fit inside the 200 ms window.
N — An individual ssdp:alive message must have all corresponding ssdp:alive sent within a 10 second transmission window. (see 7.2.4.4)
O — This delay is not drawn to scale.

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	S	R	DMP	7.2.4.7 Due to the unreliable nature of UDP, control points should send each M-SEARCH message more than once.	[28]	Wireless access points do not retry multicast traffic and may cause UPnP discovery problems. This recommendation repeats advice from UPnP DA 1.0.1.
	S	R	DMP	7.2.4.8 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.	[28]	
	S	L	DMS	7.2.4.9 Upon startup, UPnP devices should broadcast an ssdp:byebye before sending the initial ssdp:alive onto the local network.	[28]	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.
7.2.5 DDC UPnP HTTP Support and General Rules	M	R	DMS, DMP	7.2.5.1 UPnP endpoints (devices and control points) must support at least HTTP/1.0 ([19]) for performing UPnP communications, excluding SSDP communications.	[19] [28]	This guideline specifies that UPnP transactions that use HTTP must minimally support HTTP/1.0.
	M	R	DMS	7.2.5.2 UPnP devices must support HTTP/1.1.	[24] [28]	Although HTTP/1.0 is the baseline for UPnP communications, HTTP/1.1 is recommended.
	M	R	DMP	7.2.5.3 HTTP servers of UPnP control points must support HTTP/1.1.		
	S	C	DMP	7.2.5.4 HTTP clients of UPnP control points should use and support HTTP/1.1.		

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	R	DMS, DMP	7.2.5.5 The message format of HTTP responses (sent by HTTP servers of both devices and control points) must be compliant with the version number specified by the request.	[21]	The clarifying IETF specification ([21]) 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.
	S	R	DMS, DMP	7.2.5.6 HTTP/1.1 servers should return HTTP version 1.1 in the response header, regardless of the version specified in the HTTP client's request.	[21]	
	M	R	DMS, DMP	7.2.5.7 HTTP servers must not report a higher version of HTTP than is actually supported by the implementation.	[24]	
	M	R	DMS, DMP	7.2.5.8 The HTTP servers and clients of UPnP endpoints (devices and control points) must be able to properly parse all HTTP headers provided to them. In particular, they must 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 must ignore the header and continue parsing the packet.	[19] [24]	This guideline specifies a minimal robustness level for parsing HTTP headers. Under no circumstances should the UPnP endpoint fail to process a properly formed HTTP packet because of an unrecognized or unsupported field.
	M	R	DMS, DMP	7.2.5.9 The HTTP servers and clients of UPnP endpoints (devices and control points) must 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 must 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 must be encoded in UTF-8. UPnP endpoints (devices and control points) that receive a content type of *text/xml* must assume UTF-8 character set encoding.	[19] [24]	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.

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.2.6 DDC UPnP HTTP/1.0 Rules	M	C	DMS, DMP	7.2.6.1 For all HTTP/1.0 transactions, the HTTP server must immediately 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.	[19] [24]	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.
	M	R	DMS	7.2.6.2 If a UPnP device's HTTP server responds to an SOAP request as part of an HTTP/1.0 transaction, then the UPnP device must close the socket 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. It should be noted that in both of the above cases, the UPnP device has the HTTP server and the UPnP control point issues the HTTP request.	[19] [24]	This is the proper behavior for a UPnP device, as it follows standard HTTP rules.
	M	R	DMP	7.2.6.3 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 must close the socket 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. It should be noted that in both of the above cases, the UPnP control point has HTTP server and the UPnP devices issues the HTTP request.	[19] [24]	This is the proper behavior for a UPnP control point.

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.2.7 DDC UPnP HTTP/1.1 Transaction Rules	M	R	DMS	7.2.7.1 A UPnP device's HTTP server must close the socket after responding to an SOAP request with the **CONNECTION: CLOSE** token.	[24]	
	M	R	DMP	7.2.7.2 A UPnP control point's HTTP server must close the socket after responding to an event that was sent with the **CONNECTION: CLOSE** token.		
	M	R	DMS, DMP	7.2.7.3 HTTP clients of UPnP endpoints (devices and control points) must 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.	[24]	Only HTTP clients that support *Chunked Transfer Coding* and 100 (Continue Response) messages can initiate HTTP/1.1 transactions.
	M	R	DMS, DMP	7.2.7.4 The HTTP servers of UPnP endpoints (devices and control points) must use the **CONTENT-LENGTH** HTTP header tag at all times, unless the connection will be closed immediately after the response is sent or *Chunked Transfer Coding is* used.	[24]	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.
7.2.8 DDC UPnP HTTP Persistent Connections	M	R	DMS, DMP	7.2.8.1 The HTTP clients and servers of UPnP endpoints (devices and control points) must not use the **CONNECTION: KEEPALIVE** token for HTTP/1.0 transactions.	[19] [24]	Although HTTP/1.0 supports persistent connections by way of the KeepAlive token, there are interoperability issues because the methodology is an experimental extension.
	S	R	DMS, DMP	7.2.8.2 The HTTP clients and servers of UPnP endpoints (devices and control points) should support persistent HTTP/1.1 connections and pipelining.	[24]	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.
	S	R	DMS, DMP	7.2.8.3 The HTTP clients of UPnP endpoints should use persistent HTTP/1.1 connections.	[24]	It should be noted that 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.

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	L	DMP, DMS	7.2.8.4 The HTTP clients of UPnP endpoints must 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.	[24]	This guideline ensures consistent and correct behavior between mixes of UPnP endpoints that may or may not support HTTP pipelining.
	M	C	DMS, DMP	7.2.8.5 The HTTP servers of UPnP endpoints (devices and control points) that do not support persistent connections must answer the first HTTP request from the requesting UPnP control point and close the TCP connection to correctly ignore other requests.	[24]	This prevents control points and devices from holding network sockets for an unnecessarily long period.
	M	C	DMP, DMS	7.2.8.6 The HTTP clients of UPnP endpoints that send multiple requests in a single HTTP session must be ready to open new HTTP sessions if the device does not respond to all requests on the initial HTTP session.	[24]	
	M	L	DMS, DMP	7.2.8.7 The HTTP clients of UPnP endpoints must 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 transfer/streaming.	[24]	

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.2.9 DDC UPnP Device Responsiveness	M	L	DMS	7.2.9.1 UPnP devices must begin the transmission of an SOAP response within 27 seconds of receiving a complete SOAP request.	[28]	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 must still begin their response within 30 seconds. It should be noted that a UPnP AV MediaServer can reduce a long transmission time for an SOAP response (for a CDS:Browse or CDS:Search action) by reducing the number of returned items in the result. See the 7.3.9 MM CDS Browse/Search Action: Reduced Response Behavior] guideline for more information.
	S	C	DMS	7.2.9.2 UPnP devices should begin the transmission of SOAP responses as soon as possible.		
	S	C	DMS	7.2.9.3 UPnP devices should complete the transmission of an SOAP response within 29 seconds.		
	O	C	DMP	7.2.9.4 A UPnP control point may terminate the TCP connection for an SOAP response transmission that exceeds 30 seconds.		

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.2.10 DDC UPnP Device Description Rules	M	L	DMS	7.2.10.1 The total byte size of a device description file must not exceed 20,480 bytes (20 KB). This byte limit includes the HTTP headers.	[28]	Provides a known maximum size for device description documents.
	M	A	DMS	7.2.10.2 DLNA UPnP devices must 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, 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>. An example of <dlna:X_DLNADOC> element is shown as follows: • <dlna:X_DLNADOC xmlns:dlna="urn:schemas-dlna-org:device-1-0">DMS-1.00</dlna:X_DLNADOC> The "urn:schemas-dlna-org:device-1-0" namespace must be specified for the <dlna:X_DLNADOC> element and the namespace prefix must be “dlna:”. The namespace prefix declaration may be specified in the <root> element of the device description 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-version • dlna-dev-class = "DMS" | other-dev-class • other-dev-class = *<character, 'A' - 'Z'> • 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-version represents a version of Interoperability Guidelines supported by the DLNA device class.		Provides an easy way of distinguishing UPnP devices that are claimed as being DLNA devices. This guideline specifies the scoping rules for the <dlna:X_DLNADOC> element. Essentially, UPnP devices (in a device hierarchy) must 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.

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	A	DMP	7.2.10.3 DLNA device class implementations must ignore the element value of < dlna:X_DLNADOC>. For example, DLNA control point must not filter out a DLNA device because the version value is different from expected.		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 may change.

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.2.11 DDC UPnP Embedded Device Support	M	L	DMS	7.2.11.1 DLNA UPnP devices must 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.	[28]	A UPnP control point must 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.
	M	L	DMP	7.2.11.2 UPnP control points must 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.		
	M	L	DMS	7.2.11.3 DLNA UPnP devices must 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. It should be noted that the definition assumes Device-A and Device-B are in the same device hierarchy. Furthermore, Device-B may or may not be a DLNA device, as indicated by the presence of the <dlna:X_DLNADOC> element.		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.
	M	L	DMP	7.2.11.4 DLNA control points must not assume any *functional dependency* between embedded devices that do not contain the <dlna:X_DLNADOC> element.		
	O	C	DMS	7.2.11.5 UPnP AV devices may be implemented as a descendent of a UPnP root device, which may or may not be a standard UPnP device type.		

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	O	L	DMS	7.2.11.6 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 7.2.11.1.		This guideline permits that a device hierarchy to have UPnP devices that do not have the <dlna:X_DLNADOC> element
	M	L	DMS	7.2.11.7 Non-DLNA UPnP devices in a device hierarchy must not use the <dlna:X_DLNADOC> element.		
7.2.12 DDC UPnP Service Description Rules	M	C	DMS	7.2.12.1 Optional actions listed in the SCPD must be supported and not return the NOT_IMPLEMENTED UPnP error in response to an invocation. Optional actions that are not implemented must not be listed in the SCPD.	[28]	UPnP devices must 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.
	M	L	DMS	7.2.12.2 UPnP state variables must not be present unless they are actually used by the device, either as an evented state variable or as an action parameter.	[28]	This guideline is in consideration of the fact that control points may run on a platform with limited resources.
	S	L	DMS	7.2.12.3 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).	[28]	Although control points should 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.
	M	R	DMS	7.2.12.4 Services with evented state variables must support **SUBSCRIBE** and **UNSUBSCRIBE** operations.	[28]	Specifies normative behavior for services with evented state variables.
	M	R	DMS	7.2.12.5 DCP-required or SCPD-specified state variables with the attribute SendEvent="Yes" must actually be evented.		
	M	R	DMS	7.2.12.6 Service description files must not exceed 51,200 bytes (50 KB). This byte limit includes the HTTP headers.	[28]	This provides a reasonable maximum length for service description files.
7.2.13 DDC UPnP XML Namespace	M	L	DMS	7.2.13.1 Default namespace defined by the UPnP DA must be used in device and service descriptions.	[28]	All standard elements in device and service descriptions do not use a namespace prefix.

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.2.14 DDC UPnP Action Argument Encoding	M	R	DMS	7.2.14.1 UPnP devices must accept SOAP action arguments in the same order as specified in the standardized DCP. The order of arguments of SOAP actions in an SCPD must match the standardized DCP.	[28]	
7.2.15 DDC UPnP SOAP Packet Size	M	L	DMS	7.2.15.1 UPnP devices must be able to accept SOAP requests that are up to 20,480 bytes (20 KB) in size. This byte limit includes the HTTP headers.	[28]	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.
	M	L	DMP	7.2.15.2 UPnP control points must be able to accept SOAP responses that are up to 204,800 bytes (200 KB) in size. This byte limit includes the HTTP headers.		Security recommendations call out 200 KB as a reasonable upper bound for SOAP responses (total size for headers and body).
	O	L	DMP	7.2.15.3 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.		
7.2.16 DDC UPnP Error Codes	S	R	DMP, DMS	7.2.16.1 UPnP endpoints (devices and control points) are recommended to 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 may be necessary to simply close a UPnP initiated connection upon encountering an error condition.	[28]	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.
7.2.17 DDC UPnP GENA Packet Size	M	L	DMP	7.2.17.1 UPnP control points must 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.	[28]	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.
	O	L	DMP	7.2.17.2 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.		

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.2.18 DDC UPnP Subscription Handling	M	L	DMS	7.2.18.1 The **SUBSCRIBE** response must 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.	[28]	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 must 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.
	M	C	DMS	7.2.18.2 UPnP devices must assign a globally unique SID, where the global context is defined as the UPnP network. 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:00000000-0000-0000-0000000000000000	[28]	See guideline recommendation 7.2.19 DDC UPnP UUID Generation] for a way to generate a globally unique SID.
7.2.19 DDC UPnP UUID Generation	S	R	DMS	7.2.19.1 UPnP devices should use the DCE 1.1 methodology for generating a globally unique UUID value.	[72]	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 [72]
7.2.20 DDC UPnP Event Subscription Renewals	M	C	DMP	7.2.20.1 If UPnP control points want to continue receiving UPnP events, then they must renew their subscriptions before the negotiated subscription **TIMEOUT** expires.	[28]	This guideline instructs developers that the control point is responsible for renewing subscriptions in a timely manner.

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.2.21 DDC UPnP Event Notification Handling	M	R	DMS	7.2.21.1 UPnP devices must send events to all properly subscribed UPnP control points. The device must 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.	[28]	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 must 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 may incorrectly indicate to the user that a device is stalled or is malfunctioning.
	S	R	DMS	7.2.21.2 UPnP devices should monitor their subscription lists and remove control points that fail to renew their subscription within the negotiated time.	[28]	
7.2.22 DDC UPnP Unknown Header/Tag/Field Robustness Rule	M	R	DMP, DMS	7.2.22.1 UPnP endpoints (devices and control points) must 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.	[19] [24] [28]	This guideline addresses forward compatibility and also ensures broader interoperability between implementations that employ vendor extensions in the manner described by the guideline.

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.2.23 DDC URI Rules	M	L	DMS, DMP	7.2.23.1 All absolute URIs used for *UPnP communications* must 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	[28]	These guidelines are mandatory because DLNA device classes cannot depend on DNS infrastructure within a home network environment.
	M	L	DMS, DMP	7.2.23.2 The *a.b.c.d* format for IPv4 addresses must be used for *UPnP communications*, where each quad represents a byte in network byte order form.		
	M	L	DMS, DMP	7.2.23.3 HTTP URI escaping is always performed according to the URI specification ([18]) as required in section 3.2.1 of the HTTP/1.1 specification ([24]).	[18] [24]	This guideline specifies how to escape URI values.
	M	L	DMS, DMP	7.2.23.4 All URIs used for *UPnP communications* must 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 the URIs in the presentation files themselves, nor does it cover informational URIs for the manufacturer or product/model.	[28]	These guidelines provide a maximum URI length for the UPnP layer. See guideline 7.3.26 MM URI Rules sub-requirement 7.3.26.4 for the maximum URI length at the UPnP AV layer. According to section 2.10 of [6], 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.
	M	L	DMS, DMP	7.2.23.5 All URIs (not used for *UPnP communications*) must 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	[28]	

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	S	L	DMS	7.2.23.6 UPnP devices should not use the <URLBase> element in the device description document. When this element is omitted, the <URLBase> is the base path of the **LOCATION** value (URL) in the device advertisement.	[28]	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.
	O	R	DMS	7.2.23.7 If a URI in a device description or SCPD (service description) file is used for SOAP actions, GENA events, SCPD files, or UPnP presentation files, then the URI may be a relative URI with its base as the base path of the **LOCATION** URL value. The following is an example of the above guideline. • The ssdp:alive message has **LOCATION** value of http:/172.16.0.2/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 full URL for that service is as follows: • SCPDURL: http://172.16.0.2/service_desc.xml • controlURL: http://172.16.0.2/control eventSubURL: http://172.16.0.2:3000/sub		
	M	R	DMP	7.2.23.8 UPnP control points must work with UPnP devices that use a <URLBase> element and with those that do not use a <URLBase> element. Control points must also work with UPnP devices that use absolute or relative URIs.		
	M	R	DMS	7.2.23.9 UPnP devices must use the **CALLBACK** URI values sent by control points for event delivery. (I.e. a UPnP device must use **CALLBACK** URI values as exactly as received that are consistent with guidelines 7.2.23.1 – 7.2.23.4.)		Callback URI requirements help ensure interoperability by limiting edge cases.

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.2.24 DDC UPnP Device description Usage	M	C	DMS	7.2.24.1 If a DLNA UPnP device wants to change a device description or a service description files, then the UPnP device needs 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.	[28]	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.
	M	C	DMP	7.2.24.2 A DLNA UPnP control point that removes a UPnP device from its list of active devices must 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 must reload the device description and service description files.	[28]	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.
7.2.25 DDC UPnP UDN Usage	S	A	DMS	7.2.25.1 UPnP devices should not change the UDN between reboots or application launch/shutdown.	[28]	Implementing this guideline enables usages such as *my favorite devices*. Generally, UPnP device UDN values should 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).
	M	A	DMS	7.2.25.2 UPnP devices must not change the UDN if only the <friendlyName> or IP addresses values are changed, although UDN may be changed if a UPnP device changes its device description or any of its supported services.		UPnP control point can identify UPnP devices even if FriendlyName or IP addresses are changed.
	M	C	DMS	7.2.25.3 If a UPnP device UDN changes, it must re-advertise on the network using the new UDN.	[28]	Control points that receive the advertisement know that a new UPnP device is available. This is required by guideline 7.2.10, "DDC UPnP Device Description Rules".

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	C	DMS	7.2.25.4 If a UPnP device UDN changes, it must send an ssdp:byebye for the old UDN.	[28]	Without this guideline, UPnP control points will have no idea that the old UDN is no longer valid. This is required by guideline 7.2.10, "DDC UPnP Device Description Rules".
	M	C	DMS	7.2.25.5 A UPnP device must limit their UDN to a UTF-8 encoded string value containing "uuid:" followed by a UUID, which is a globally unique 128-bit value in hexadecimal form, with a maximum byte length of 68 bytes, including the "uuid:" text. Dash (-) characters are permitted between hex values. Example: uuid:00000000-0000-0000-0000000000000000	[28]	See guideline recommendation 7.2.19, "DDC UPnP UUID Generation" for a way to generate a globally unique 128-bit value for the UDN.
7.2.26 DDC UPnP Multi Homing Rules	O	C	DMS	7.2.26.1 When a UPnP device has multiple IP addresses, the device may advertise on those IP addresses with the same or different UDN.	[28]	Multiple home network segments, wireless networking, and Auto IP can combine to create usability problems that can be avoided by following the specified rules.
	M	C	DMS	7.2.26.2 Each UPnP device advertisement must contain a return IP address of the home network interface it is sent on.	[28]	
	S	C	DMP	7.2.26.3 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.	[28]	
	O	L	DMP	7.2.26.4 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.	[28]	

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.2.27 DDC UPnP Device Icons	M	L	DMS	7.2.27.1 If a UPnP device provides a device icon, the UPnP device must provide two JPEG icons that conform to the 7.6.7 JPEG_SM_ICO and 7.6.8 JPEG_LRG_ICO media format profiles and two PNG icons that conform to the 7.6.9 PNG_SM_ICO and 7.6.10 PNG_LRG_ICO media format profiles.	[28]	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.
	O	R	DMS	7.2.27.2 UPnP devices may provide additional icons in other formats besides PNG and JPEG.	[28]	
	M	R	DMS	7.2.27.3 UPnP devices must 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 must conform to the DLNA icon media format profiles 7.6.7 JPEG_SM_ICON, 7.6.8 JPEG_LRG_ICON, 7.6.9 PNG_SM_ICON, and 7.6.10 PNG_LRG_ICON respectively. The following values are recommended for the <depth> element which indicates color bits per pixel for PNG and JPEG device icons.]	[28]	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 recommends the values for <depth> element required by the UPnP DA. Note that the values for PNG do not help to identify color types.

Chart 7-1 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 MUST BE 8 BITS)	24
JPEG	(8 BITS Y/CR/CB SAMPLES)	24

Table 7-2 Device Discovery and Control Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.2.28 DDC UPnP UTF-8 Support	M	L	DMS, DMP	7.2.28.1 UPnP endpoints (devices and control points) must use UTF-8 encoding of all XML fragments. UPnP endpoints must be tolerant of the UTF-8 maximum of 4 bytes of Unicode character as required by XML processors.	[6] [22] [28]	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 must decode up to 4 bytes of character. Specifically, see section 2.2 of [6] for more information. It calls out any *Unicode character, excluding the surrogate blocks, FFFE, and FFFF.*
7.2.29 DDC UPnP XML Comments	M	L	DMS, DMP	7.2.29.1 UPnP endpoints (devices and control points) must never source XML with comments.	[28]	XML comments normally have to be skipped by XML parsers. This guideline ensures that comments do not prevent interoperation.
	O	C	DMS, DMP	7.2.29.2 UPnP endpoints (devices and control points) may reject any XML provided with comments.		
7.2.30 DDC UPnP Boolean Types	M	L	DMS, DMP	7.2.30.1 UPnP endpoints (devices and control points) must use “0” for false and “1” for true when using the UPnP Boolean type.	[28]	This simplifies control point implementations and also reduces the size of some UPnP traffic.

7.3 MEDIA MANAGEMENT

This section of the DLNA Home Networked Device Interoperability Guidelines covers the guidelines for implementing media management using the UPnP AV architecture.

Table 7-3 Media Management Guidelines

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
				General Capability Requirements		
7.3.1 MM UPnP AV 1.0 Compliance	M	R	DMS, DMP	7.3.1.1 DLNA device classes must fully support the mandatory portions of the UPnP AV v1.0 specifications.	[29] [30] [31] [33]	UPnP AV 1.0 is the baseline architecture for discovering media sources and sinks.
				Device Requirements		
7.3.2 MM DMP UPnP AV MediaServer Control Point Definition	M	R	DMP	7.3.2.1 A DMP must implement a UPnP AV MediaServer control point for finding content on a DMS.	[29] [30] [31] [33]	This guideline indicates that a DMP device class must use a UPnP control point that controls a UPnP AV MediaServer for enumerating content.
7.3.3 MM DMS UPnP AV MediaServer Device Definition	M	R	DMS	7.3.3.1 A DMS must implement a UPnP AV MediaServer device that must have a ContentDirectory service and a ConnectionManager service.	[33]	DMS devices must implement the minimum baseline services for a UPnP AV MediaServer.
	O	R	DMS	7.3.3.2 A DMS may have an AVTransport service in the UPnP AV MediaServer device.	[33]	This guideline permits a DMS to implement the AVTransport service.
7.3.4 MM DMS ContentDirectory Rules	M	R	DMS	7.3.4.1 A DMS must support the mandatory actions and state variable for a ContentDirectory service.	[31]	
7.3.5 MM DMS ConnectionManager Rules	M	R	DMS	7.3.5.1 A DMS must support the mandatory actions and state variables for a ConnectionManager service.	[30]	This guideline specifies the minimum requirements for a DMS's ConnectionManager service.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	L	DMS	7.3.5.2 If a DMS only supports HTTP as the Media Transport, the DMS must not implement CMS:PrepareForConnection and must not implement CMS:ConnectionComplete in the UPnP AV MediaServer device. It should be noted that DMS must support HTTP/1.1 and HTTP/1.0, as described in 7.8.12	[30]	DMS devices that only support HTTP transport protocols will neither implement the ConnectionManager actions CMS:PrepareForConnection nor CMS:ConnectionComplete.
7.3.6 MM UPnP AV MediaServer Control Point Tolerance of Unknown Property	M	C	DMP	7.3.6.1 UPnP AV control points must be tolerant of all properties that appear in a DIDL-Lite XML fragment. Properties are as defined in Appendix B of[31]. Tolerant behavior is defined as being able to successfully "parse and interpret" or "parse and ignore" the DIDL-Lite XML elements and attributes.	[31]	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.
7.3.7 MM CMS:GetProtocolInfo Rules	M	L	DMS	7.3.7.1 The ConnectionManager service must list the union set of supported protocolInfo values supported by the device.	[30]	This guideline makes it easier for control points to find servers that provide content for a given profile.
	M	L	DMS	7.3.7.2 The sets of protocolInfo values returned in CMS:GetProtocolInfo must list the protocolInfo values that use the *http-get* media transport and DLNA media format profiles first.	[30]	Although the number of returned protocolInfo values may be large, control points may use this guideline to capture the subset of relevant DLNA protocolInfo values.
	M	A	DMS	7.3.7.3 The fourth field of the protocolInfo values (obtained from the ConnectionManager service) that have the DLNA.ORG_PN parameter in the fourth field must follow one of the formatting conventions. • Provide all parameters as defined in guideline 7.3.11.1 (MM DIDL-Lite protocolInfo values: 4th Field). • Provide only the DLNA.ORG_PN parameter, as defined in guideline 7.3.11.3 (MM DIDL-Lite protocolInfo values: 4th Field).	[30]	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.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.3.8 MM CDS Browse/Search Action: Filter Argument	M	C	DMS	7.3.8.1 The following five metadata properties must 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	[31] [33]	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.
	M	R	DMS	7.3.8.2 If an element of metadata property is specified, then the required attributes of the metadata element must be presented. For example: • If a control point specifies "res" in the Filter, then res@protocolInfo is returned.	[31]	
	S	R	DMP	7.3.8.3 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.	[31] [33]	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.
	M	R	DMS	7.3.8.4 In conjunction with 7.3.8.1, a UPnP AV MediaServer device must 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 an attribute property in the *Filter* automatically infers returning the element. For example: • If the control point specifies res@importURI (without "res"), then the "res" is also returned.	[31] [33]	

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.3.9 MM CDS Browse/Search Action: Reduced Response Behavior	O	C	DMS	7.3.9.1 A UPnP AV MediaServer device may reduce the number of CDS media objects (<item> and <container> elements) in a response to a CDS:Browse or CDS:Search to avoid the following scenarios. • The transmission of an SOAP response with a huge byte length (>204,800 bytes). • The transmission of an SOAP response that exceeds 30 seconds for the transmission time.	[31] [33]	This guideline allows a UPnP AV MediaServer to limit the number of media objects returned in the SOAP response, even if the control point specified a desire for more media objects in the *RequestedCount* input argument. The reason for permitting such behavior is to allow UPnP AV MediaServer implementations to comply with other guidelines: 7.2.15, "DDC UPnP SOAP Packet Size" and 7.2.9, "DDC UPnP Device Responsiveness".
	M	C	DMS	7.3.9.2 The number of CDS media object entries (total <item> and <container> elements) in the *Result* output argument (containing the DIDL-Lite metadata) must match the value specified in the *NumberReturned* output argument.	[31] [33]	This guideline must be followed, even if a UPnP AV MediaServer reduces the number of media objects returned in the SOAP response.
	M	C	DMS	7.3.9.3 If a UPnP AV MediaServer device reduces the number of CDS media objects in a CDS:Browse(BrowseDirectChildren) or CDS:Search response then the number of returned CDS objects (as parsed in *Result*) must be equal to the value of *NumberReturned*, which is less than *RequestedCount*.	[31] [33]	A UPnP AV MediaServer that limits the number of CDS media objects is obligated to return a *NumberReturned* value that is consistent with the *RequestedCount* input argument.
	M	L	DMP	7.3.9.4 A UPnP AV MediaServer control point must use a *RequestedCount* of 0 or 1 and *StartingIndex* of 0 when using CDS:Browse with *BrowseMetadata* option.	[31] [33]	Improves expectations for CDS:Browse scenarios with *BrowseMetadata*.
	M	C	DMS	7.3.9.5 A UPnP AV MediaServer device must always return one CDS object (as indicated in *RequestedCount* and *Result*) when successfully responding to a CDS:Browse request with the *BrowseMetadata* option, regardless of the *RequestedCount* value.	[31] [33]	

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	C	DMS	7.3.9.6 If the UPnP AV MediaServer device returns more than zero media 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 must be set to zero.	[31] [33]	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 media objects. Although some UPnP AV MediaServer implementations may choose to report the accurate *TotalMatches* value, at the expense of violating the 30-second timeout rule, DLNA does not recommend that implementation option. The 7.2.9, "DDC UPnP Device Responsiveness" guideline indicates that a control point is allowed to terminate an SOAP response that exceeds a 30-second transmission time.
	S	C	DMS	7.3.9.7 If a UPnP AV MediaServer device cannot find more than zero media objects (in 27 seconds, as described in 7.2.9.1), 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 an SOAP error response code of 720 (Cannot process the request).	[31] [33]	This guideline covers the scenario where a UPnP AV MediaServer can neither find any media 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 the previous guideline.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	S	C	DMP	7.3.9.8 A UPnP AV MediaServer control point should specify the desired number of media objects in the *RequestedCount* input argument of a CDS:Browse or CDS:Search query.	[31] [33]	This guideline recommends control points to request a reasonable number of media objects in a single CDS query. The number of media objects that can be displayed to the user at a single time is a good measure of reasonableness. Using a *RequestedCount* of zero may cause the transmission of a huge SOAP response, which is undesirable.
	S	C	DMP	7.3.9.9 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.	[31] [33]	Generally speaking, control points that specify smaller *RequestedCount* values will receive the response from the device sooner than if a larger value were specified.
7.3.10 MM DIDL-Lite protocolInfo values	M	A	DMS	7.3.10.1 The res@protocolInfo values obtained from a ContentDirectory service must use the rules described in guideline rows below.	[31]	This guideline provides requirements on the protocolInfo values in DIDL-Lite documents.
	M	R	DMS	7.3.10.2 If the <res> value contains an HTTP URL, then the first field of the res@protocolInfo value must be as shown below. • http-get	[31]	Reiterates the protocol string value for DLNA content transported across HTTP.
	M	L	DMS	7.3.10.3 The third field of the res@protocolInfo value must use the MIME types specified in Chart 7-5 for DLNA-normative media format profiles.	[31]	
	M	L	DMP	7.3.10.4 UPnP AV MediaServer control points must use the DLNA.ORG_PN parameter in the 4th field (7.3.11.3) of protocolInfo to identify content conforming to DLNA media format profile.	[31]	MIME-types in the 3rd field of res@protocolInfo alone are not sufficient to identify DLNA content. MIME-types are not be used to identify DLNA media format profiles because MIME-types will be updated by IANA.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.3.11 MM DIDL-Lite protocolInfo values: 4th Field	M	A	DMS	7.3.11.1 If a protocolInfo value has the first field value of "http-get", then the following definition must be used for the fourth field. • 4th_field = [pn-param] [op-param] [ps-param] [*(other-param)] The relative order of pn-param, op-param, ps-param, and *(other-param) used in 4th_field is mandatory. For example, pn-param cannot appear after op-param, ps-param, or *(other-params). The syntax and definition of pn-param, op-param, ps-param, and *(other-param) are defined in the guidelines below.	[30]	This guideline defines the syntax of the fourth field for a protocolInfo value that indicates content that is transported across HTTP.
	O	A	DMS	7.3.11.2 If a protocolInfo value has the first field value not equal to "http-get", then the fourth field may have a syntax that differs from 7.3.11.1.	[30]	This guideline permits a different syntax for the fourth field of a protocolInfo value when the content is not transported over HTTP.
	M	A	DMS	7.3.11.3 The definition of pn-param is as follows: • pn-param = "DLNA.ORG_PN=" pn-value • pn-value = *<'a'-'z', 'A'-'Z', '0'-'9', '_'> The pn-value must identify the DLNA media format profile ID that the content conforms to. The pn-param is reserved for use with content conforming to a DLNA media format profile. Use of pn-param for content not conformant with a DLNA media format profile is expressly prohibited.	[30]	This guideline defines the DLNA.ORG_PN parameter. This parameter is used to identify DLNA content. This parameter cannot be used for non-DLNA content.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	A	DMS	7.3.11.4 The definition of op-param is 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, with the semantics described below. • a-val: indicates support of the **TimeSeekRange.DLNA.ORG** HTTP header (see 7.8.24) for the content • b-val: indicates support of the **RANGE** HTTP header (see 7.8.22) for the content If the first field of protocolInfo is not "http-get" then the fourth field must omit the op-param. If the transport does not support either time-based seek operations or byte-based-seek operations for this content, then the 4th field must omit the op-param. If using the 4th_field syntax defined in 7.3.11.1 and if op-param follows another parameter, then the op-param must include the op-param-delim.	[30]	This guideline defines the DLNA.ORG_OP parameter. This parameter is used to identify support of advanced transport operations. This parameter is currently only valid for content transported over HTTP.
	O	A	DMS	7.3.11.5 In conjunction with the rules defined in 7.3.11.4, the fourth field of a protocolInfo may use the op-param in the following ways. If the content's transport server supports either time-based seek operation or byte-based-seek operation for this content then the fourth field may include the op-param. • The fourth field may include the op-param for content that is conformant to a DLNA media format profile. • The fourth field may include the op-param for content that is not conformant to a DLNA media format profile.	[30]	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.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	A	DMS	7.3.11.6 The definition of ps-param is 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 must be a comma-delimited list of play speed values that the server can support for the content. The ps-value must exclude the play speed of "1" from its list. If the media transport server does not support additional server-side play speeds beyond "1", then the fourth field must omit the ps-param (i.e. "DLNA.ORG_PS=1" is prohibited). The format of each play speed value must conform to the TransportPlaySpeed string, as specified in section 2.2.8 of[29]. If using the 4th_field syntax defined in 7.3.11.1 and if ps-param follows another parameter, then the ps-param must include the ps-param-delim.	[30]	This guideline defines the DLNA.ORG_PS parameter. The parameter indicates the transport layer's supported play speeds.
	O	A	DMS	7.3.11.7 In conjunction with the rules defined in 7.3.11.6, the fourth field of a protocolInfo may use ps-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". • The first field of protocolInfo is not "http-get".	[30]	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 non-HTTP transports.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	A	DMS	7.3.11.8 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', '_', ',', '+', '-'> If using the 4th_field syntax defined in 7.3.11.1 and if other-param follows another parameter, then the other-param must include the other-param-delim.	[30]	This defines the syntax for vendor extensions in the fourth field of protocolInfo values.
	O	A	DMS	7.3.11.9 Vendors may use other-param for vendor-specific parameters in the fourth field of a protocolInfo value.	[30]	This guideline permits the use of vendor extensions in the fourth field.
	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					
7.3.12 MM DIDL-Lite Non-empty Metadata Values	M	L	DMS	7.3.12.1 A UPnP AV MediaServer device must 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. Term applies only to guideline entries of 7.3.12.	[31]	UPnP AV MediaServers that expose DIDL-Lite metadata with empty values or values composed entirely of white-spaces can cause problems for control points. More importantly, a user has little idea on how to interpret such values.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	L	DMS	7.3.12.2 If a value of a *metadata property* is unknown, the UPnP AV MediaServer must not use the corresponding DIDL-Lite attribute or element, unless explicitly required by the DIDL-Lite schema.		
	M	C	DMP	7.3.12.3 A UPnP AV MediaServer control point must be tolerant of DIDL-Lite attributes and elements (whether defined by DIDL-Lite schema or not) that have empty values.		
7.3.13 MM DIDL-Lite Recommended Metadata Properties	M	A	DMS	7.3.13.1 Content that conforms to the DLNA "Image" media class must use **object.item.imageItem** or a derived class for the upnp:class value. Content that conforms to the DLNA "Audio only" media class must use **object.item.audioItem** or a derived class for the upnp:class value. Content that conforms to the DLNA "Video with audio (AV)" media class must 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 7.6.5 and 7.6.6)..	[31]	
	M	L	DMS	7.3.13.2 Metadata properties defined in Appendix B of the ContentDirectory Service specification ([31]) must use the following namespace prefixes: Chart 7-2 **Namespace Prefix** Namespace: DIDL-LITE — Prefix: <EMPTY STRING> Namespace: DUBLIN CORE — Prefix: DC: Namespace: UPNP U — Prefix: PNP:	[31]	

Table 7-3 Media Management Guidelines (Continued)

<table>
<tr><th>Name</th><th>M/S/O</th><th>A/C/F/L/R</th><th>Dev Class</th><th>Description</th><th>Ref #</th><th>Comments</th></tr>
<tr><td></td><td>S</td><td>A</td><td>DMS</td><td>7.3.13.3 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.
Chart 7-3 Recommended Metadata Properties
<table><tr><th>upnp:class value</th><th>Property Names</th></tr><tr><td>OBJECT.ITEM.AUDIOITEM</td><td>DC:CREATOR, UPNP:ALBUM, UPNP:GENRE, RES@DURATION</td></tr><tr><td>OBJECT.ITEM.IMAGEITEM</td><td>DC:DATE, RES@RESOLUTION</td></tr><tr><td>OBJECT.ITEM.VIDEOITEM</td><td>DC:DATE, UPNP:GENRE, RES@DURATION</td></tr><tr><td>OBJECT.CONTAINER.ALBUM.MUSICALBUM</td><td>DC:CREATOR, UPNP:GENRE</td></tr><tr><td>OBJECT.ITEM.VIDEOITEM.VIDEOBROADCAST
OBJECT.ITEM.AUDIOITEM.AUDIOBROADCAST</td><td>UPNP:GENRE, UPNP:CHANNELNAME, UPNP:CHANNELNR (APPLICABILITY OF UPNP:CHANNELNR DEPENDS ON REGION)</td></tr></table>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 may 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.</td><td>[31]</td><td>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.</td></tr>
</table>

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.3.14 MM DIDL-Lite dc:date Format	M	C	DMS	7.3.14.1 The syntax for the DIDL-Lite <dc:date> element value must conform to the following subset profile of[27]. CCYY-MM-DD(Thh:mm:ss(.sss)?([Z|([+|-]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 must have no characters for the <time-offset> part of the date grammar.	[27]	
7.3.15 MM DIDL-Lite Multiple Res: Formats	M	R	DMS	7.3.15.1 A media object (identified through an <item> or <container> element) that has multiple content binaries in different media format profiles must be exposed as multiple <res> elements. This technique can apply to the case where the original resource is transcoded dynamically.	[31]	These guidelines require a UPnP AV MediaServer device to expose multiple <res> elements for a single media object. This allows a UPnP AV MediaServer control point to present a single media object to the user, without necessarily presenting each variant of the same content.
7.3.16 MM DIDL-Lite Multiple Res: Transports	M	R	DMS	7.3.16.1 A media object (identified through an <item> or <container> element) that has multiple content binaries in different media transport protocols must be exposed as multiple <res> elements.		

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.3.17 MM DIDL-Lite Multiple Res: Thumbnails	S	C	DMS	7.3.17.1 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.)	[31]	UPnP AV MediaServer devices that implement thumbnail support reduce the network load for themselves and for control points that display thumbnails to the user.
	S	C	DMS	7.3.17.2 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.)	[31]	
	M	L	DMS	7.3.17.3 If a UPnP AV MediaServer exposes thumbnail images for image or video content, then a UPnP AV MediaServer must provide a thumbnail that conforms to requirement 7.6.5 JPEG_TN media format profile and be declared with the JPEG_TN designation in the fourth field of the res@protocolInfo attribute.	[31]	When thumbnails are provided, the minimal expectation is to provide JPEG thumbnails. However, vendors may also provide additional thumbnails of other formats (such as a PNG thumbnail).
	O	C	DMS	7.3.17.4 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.	[31]	

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.3.18 MM DIDL-Lite AudioItem Album Art	S	C	DMS	7.3.18.1 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.	[31]	Unlike image or video content, thumbnails for audio content should be presented through the <upnp:albumArtURI> element.
	S	C	DMS	7.3.18.2 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 7.6.5 for the JPEG_TN media format profile.	[31]	If album art thumbnails are provided, the desired expectation is to have JPEG thumbnails. Additional thumbnails can also be provided.
	M	A	DMS	7.3.18.3 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> must have the albumArtURI@dlna:profileID attribute that identifies the DLNA profile ID of the thumbnail. The namespace for DLNA defined properties must be “urn:schemas-dlna-org:metadata-1-0/” and the namespace prefix must be “dlna:”. The following is an 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>	[31]	This guideline allows control points a convenient way to identify thumbnails that conform to a DLNA media format profile.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.3.19 MM DIDL-Lite Content: Multiple Points of Accessibility	M	L	DMS	7.3.19.1 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) must 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 7.3.26, "MM URI Rules". 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.	[31]	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.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
						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). Section Appendix C (Informative), 'UPnP Devices with Multiple Network Interfaces' describes the subtleties of multiple network interfaces and the role of these guidelines in more detail

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	A	DMS	7.3.19.2 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) must 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 must expose all URI values either through multiple <res> elements (for each CDS object) or multiple CDS objects. Please see guidelines 7.3.19.3 and 7.3.19.4 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 7.3.26, "MM URI Rules". 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.	[31]	
	S	C	DMS	7.3.19.3 In conjunction with guideline 7.3.19.2, 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.	[31]	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 7.3.19.2.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	O	C	DMS	7.3.19.4 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. It should be noted that it is generally true that a UPnP AV MediaServer may return CDS objects with zero or more <res> elements.	[31]	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 7.3.19.1.
7.3.20 MM DIDL-Lite Objects for Multipage Images	S	C	DMS	7.3.20.1 If a UPnP AV MediaServer exposes an image file that contains multiple pages of images (such as a multi-page TIFF file), then the UPnP AV MediaServer should use a CDS container object (<upnp:class> value of or derived from **object.container**) to represent the image file.	[31]	This guideline recommends a normative behavior for how a UPnP AV MediaServer can advertise multi-page image files. If the recommended technique of using a container object is used, then there are mandatory elements that must be implemented as part of the recommended technique.
	M	C	DMS	7.3.20.2 If a UPnP AV MediaServer exposes an image file that contains multiple pages of images and the UPnP AV MediaServer uses a CDS container object to represent the image file, then the UPnP AV MediaServer must contain item objects (<upnp:class> values of or derived from **object.item.imageItem**) that represent the individual pages of the image file. The order of the child items must match the order of the corresponding pages in the image file.	[31]	
	M	C	DMS	7.3.20.3 If a UPnP AV MediaServer exposes an image file that contains multiple pages of images and the UPnP AV MediaServer uses a CDS container object to represent the image file, then the UPnP AV MediaServer must supply a <res> element for the image file (that contains the multiple pages) as a <res> element of the corresponding CDS container object.	[31]	

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.3.21 MM DIDL-Lite Boolean Values	M	L	DMS	7.3.21.1 DIDL-Lite Boolean values must use “0” for false and “1” for true.	[31]	This simplifies control point implementations and also reduces the size of some UPnP traffic.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.3.22 MM DIDL-Lite Desc Element Use	M	R	DMS	7.3.22.1 The text in the <desc> element must be a valid XML fragment with declared namespaces. Shown below are some examples of proper <desc> usage. • <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 xmlns:example="http://some.example.org/yada/”> <example:yada>some desc data</example:yada> </desc> </container> </DIDL-Lite> • <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:example="http://some.example.org/yada/”"> <container id="0000000000000016" searchable="1" parentID="000000000000000A" restricted="0" childCount="1"> <dc:title>some title</dc:title> <upnp:class>object.item</upnp:class> <desc> <example:yada>some desc data</example:yada> </desc> </container> </DIDL-Lite>	[31]	The XML contents of the <desc> are not validated by a DIDL- Lite validation tool, and the validity of the contained XML block is out of scope with respect to this document. Vendors are encouraged to utilize only valid fragments in the <desc> block.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.3.23 MM DIDL-Lite Restrictions	M	L	DMS	7.3.23.1 UPnP AV endpoints (devices and control points) must never employ 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 must never contain XML comments. UPnP AV endpoints may reject any XML that is not encoded with these restrictions.	[31]	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.
7.3.24 MM DIDL-Lite Max Metadata Length	M	L	DMS	7.3.24.1 Unless specified in another DLNA guideline, element values and attribute values, appearing in DIDL-Lite documents, that are *length-unlimited* must 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.	[31]	This guideline puts a worst-case limit on all other metadata values in CDS. This allows for smaller limits to be specified, but that at this time this is a true maximum.

Table 7-3 Media Management Guidelines (Continued)

<table>
<tr><th>Name</th><th>M/S/O</th><th>A/C/F/L/R</th><th>Dev Class</th><th>Description</th><th>Ref #</th><th>Comments</th></tr>
<tr><td></td><td>M</td><td>L</td><td>DMS</td><td>7.3.24.2 In DIDL-Lite documents, element and attribute values that are length-limited must 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>
<tr><th>Data Type in CDS</th><th>Data Type in UPnP Device Architecture</th><th>Maximum Byte Length</th></tr>
<tr><td>BOOLEAN</td><td>BOOLEAN</td><td>5</td></tr>
<tr><td>UNSIGNED INTEGER</td><td>UI4</td><td>10</td></tr>
<tr><td>INTEGER</td><td>I4</td><td>11</td></tr>
<tr><td>UNSIGNED LONG</td><td>N/A</td><td>20</td></tr>
<tr><td>LONG</td><td>N/A</td><td>21</td></tr>
<tr><td>N/A</td><td>UI1</td><td>3</td></tr>
<tr><td>N/A</td><td>UI2</td><td>5</td></tr>
<tr><td>N/A</td><td>I1</td><td>4</td></tr>
<tr><td>N/A</td><td>I2</td><td>6</td></tr>
<tr><td>N/A</td><td>INT</td><td>11</td></tr>
<tr><td>N/A</td><td>R4</td><td>14</td></tr>
<tr><td>N/A</td><td>R8</td><td>22</td></tr>
<tr><td>N/A</td><td>NUMBER</td><td>22</td></tr>
<tr><td>N/A</td><td>FIXED.14.4</td><td>20</td></tr>
<tr><td>N/A</td><td>FLOAT</td><td>110*1</td></tr>
<tr><td>N/A</td><td>CHAR</td><td>1</td></tr>
<tr><td>N/A</td><td>DATE</td><td>10</td></tr>
<tr><td>N/A</td><td>DATETIME</td><td>19</td></tr>
<tr><td>N/A</td><td>DATETIME.TZ</td><td>29</td></tr>
<tr><td>N/A</td><td>TIME</td><td>8</td></tr>
<tr><td>N/A</td><td>TIME.TZ</td><td>18</td></tr>
</table>
*float must be in the canonical representation.</td><td>[31]</td><td>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 7.2.30 restricts using Boolean values to “0” and “1” for DLNA device classes, it needs to be tolerant that “true” or “false” may be encountered for non DLNA devices. Hence the reason for setting the maximum length to 5.</td></tr>
</table>

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	S	L	DMS	7.3.24.3 The dc:title metadata property should not exceed 256 bytes in the XML-escaped form encoded in UTF-8.	[31]	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 7.3.33, "MM Tuner Properties: Channel Title" with foreign languages.
	M	L	DMS	7.3.24.4 The following metadata properties must not exceed 256 bytes each in the XML-escaped form encoded in UTF-8. • upnp:class • Any *length-unlimited* metadata property in Chart 7-3 • All *length-unlimited* DIDL-Lite schema defined attributes for <res>, except res@importURI. (The length for res@importURI is governed by guideline 7.3.26.)	[31]	This guideline provides control points with some information about how much memory will be needed to represent a media object.
7.3.25 MM ObjectID Usage	M	R	DMS	7.3.25.1 UPnP AV MediaServers must 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.	[31]	This is a requirement of the CDS specification.
	M	L	DMS	7.3.25.2 A UTF-8 encoded object ID must not exceed 256 bytes in the XML escaped form encoded in UTF-8.	[31]	Provides a reasonable maximum length for objectID values, which are essential for media object declarations.
7.3.26 MM URI Rules	M	L	DMS	7.3.26.1 If the metadata of a <res> element indicates that a content binary conforms to a DLNA media format profile, then the URI value of the <res> element must be an absolute URI, with the IP address in the *a.b.c.d* IPv4 address format (i.e. quad-form network byte order).	[33]	This guideline mandates that content URIs (discovered through a UPnP AV MediaServer ContentDirectory service) must use absolute URIs that use IP addresses.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	L	DMS	7.3.26.2 UPnP AV MediaServer must only advertise URIs that is properly URI-escaped in a UTF-8 encoded form.	[18] [31]	This guideline requires a UPnP AV MediaServer to advertise 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.
				7.3.26.3 HTTP URI escaping must be performed according to the URI specification ([18]) as required in section 3.2.1 of the HTTP/1.1 specification ([24]).	[18] [24]	
	M	L	DMS	7.3.26.4 URI values that appear in DIDL-Lite must not exceed 1024 bytes, in the URI-escaped UTF-8 encoded form. URI values must not have preceding or trailing white space characters.	[6] [18] [31]	URI values are theoretically infinite in length. This guideline puts a reasonable limit on the length of advertised content URI values.
	O	L	DMS	7.3.26.5 If the metadata of a <res> element indicates that a content binary does not conform to a DLNA media format profile, then the URI value of the <res> element may be a URI, with a domain name.	[33]	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.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.3.27 MM upnp:class Values	M	R	DMP	7.3.27.1 UPnP AV MediaServer control point must minimally treat derived classes in the same way as its ancestor class(es).	[31]	As an example, a UPnP AV MediaServer control point needs to be able to recognize a media 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 may be interested in certain types of content classes.
	M	R	DMP	7.3.27.2 A UPnP AV MediaServer control point must be tolerant ("parse and interpret" or "parse and ignore") of unknown upnp:class values.	[31]	Tolerance of unknown values is required, regardless of whether the control point intends to show the CDS objects to a user.
7.3.28 MM Search Capabilities	S	L	DMS	7.3.28.1 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	[31]	This requirement describes the recommended search capabilities for a UPnP AV MediaServer that supports search operations.
	M	R	DMS	7.3.28.2 All searchable properties must be listed in the return value of CDS:GetSearchCapabilities if the device implements CDS:Search.	[31]	This guideline mandates that supported search properties be discoverable.

Table 7-3 Media Management Guidelines (Continued)

<table>
<tr><th>Name</th><th>M/S/O</th><th>A/C/F/L/R</th><th>Dev Class</th><th>Description</th><th>Ref #</th><th>Comments</th></tr>
<tr><td></td><td>M</td><td>L</td><td>DMS</td><td>7.3.28.3 If a UPnP AV MediaServer supports CDS:Search, then it must minimally support the following operators for the specified property types.
Chart 7-4 Property Operators
<table><tr><th>Property</th><th>Operators</th></tr><tr><td>@ID</td><td>=</td></tr><tr><td>@REFID</td><td>= , EXISTS</td></tr><tr><td>UPNP:CLASS</td><td>DERIVEDFROM</td></tr><tr><td>ANY DATE, TIME, DURATION-BASED PROPERTY TYPES</td><td>< , <= , >= , > , = , !=</td></tr><tr><td>ALL OTHER STRING-BASED PROPERTY TYPES OTHER THAN THOSE LISTED IN PREVIOUS TABLE ENTRIES</td><td>CONTAINS , =</td></tr><tr><td>ALL URI-BASED PROPERTY TYPES</td><td>CONTAINS , =</td></tr><tr><td>INTEGER OR NUMERICAL PROPERTY TYPES</td><td>< , <= , >= , > , = , !=</td></tr><tr><td>BOOLEAN-BASED PROPERTY TYPES</td><td>= , !=</td></tr><tr><td>ANY ATTRIBUTE OR ELEMENT</td><td>EXISTS</td></tr></table>Vendors are free to apply additional CDS-normative operators for these properties or property types.</td><td>[31]</td><td>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.</td></tr>
<tr><td></td><td>M</td><td>L</td><td>DMS</td><td>7.3.28.4 If a UPnP AV MediaServer reports a searchable property for a particular data type, then a UPnP AV MediaServer must implement the associated operators for that data type.</td><td>[31]</td><td>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.</td></tr>
</table>

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.3.29 MM Container Update IDs Event	S	L	DMS	7.3.29.1 UPnP AV MediaServer devices should implement behavior for the *CDS.ContainerUpdateIDs* state variable.	[31]	This guideline is a benefit to both devices and control points, although lightweight UPnP AV MediaServer ContentDirectory Service (CDS) implementations may 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 must 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.
7.3.30 MM Tuner Container	S	A	DMS	7.3.30.1 A tuner should be represented as a container with a class of **object.container** or any derived class. A tuner container should have an associated name. The name is given by property dc:title.	[31]	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, "Tuner Representation" for recommendations on how to represent a turner container.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	A	DMS	7.3.30.2 A tuner container must have a property <:dlna:containerType> and must have a value of **Tuner_1_0**. The name space for DLNA defined properties must be "urn:schemas-dlna-org:metadata-1-0/" and the namespace prefix must be "dlna:". The following is an example. <dlna:containerType xmlns:dlna="urn:schemas-dlna-org:metadata-1-0/"> Tuner_1_0 </dlna:containerType>	[31]	
	M	A	DMS	7.3.30.3 A tuner container must contain object items of class **object.item.videoItem.videoBroadcast** or **object.item.audioItem.audioBroadcast** or both. (Objects derived from either class also qualify.)	[31]	
	S	A	DMS	7.3.30.4 The order of the object items (order of <item> elements in a CDS:Browse response)in a tuner container should correspond to the tuner up/down operation. For example, channel number or channel preset order.	[31]	
7.3.31 MM Audio Tuner	S	A	DMS	7.3.31.1 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.	[31]	These guidelines allow control points to identify content sourced from a tuner.
7.3.32 MM Video Tuner	S	A	DMS	7.3.32.1 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.	[31]	
7.3.33 MM Tuner Properties: Channel Title	S	C	DMS	7.3.33.1 The dc:title should describe the program content if available otherwise should contain the contents of the channelName CDS property. If channelName CDS property is not available the upnp:channelNr property should be used.	[31]	This is to clarify the meaning of title in context of a tuner. Some vendors may interpret title as channel name.
7.3.34 MM Tuner Properties: Channel Number	S	C	DMS	7.3.34.1 The broadcast object item should have the associated property upnp:channelNr.	[31]	This guideline clarifies the intended usage of upnp:channelNr.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	C	DMS	7.3.34.2 If upnp:channelNr is used, then each upnp:channelNr number must be unique within the context of its container.	[31]	
7.3.35 MM Tuner Properties: Channel Name	S	A	DMS	7.3.35.1 If the broadcast object item has the associated property upnp:channelName, then each upnp:channelName string should be unique within the context of its container. The channelName should be used to identify the channels, not the program content.	[31]	This guideline clarifies the intended usage of upnp:channelName.
7.3.36 MM Tuner Content URI	M	A	DMP	7.3.36.1 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.	[31]	This guideline essentially requires tuner content to be advertised and accessible through a URI.
7.3.37 MM Tuner Control Point Assumptions	S	A	DMP	7.3.37.1 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 may change without the client being aware of the change.	[31]	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 rendering endpoints connected as clients or if the tuner is being used by a local output device.
7.3.38 MM IFO File UR	O	A	DMS	7.3.38.1 If a UPnP AV MediaServer exposes AV media content profiled according to 7.7.12, MPEG_PS_NTSC profile or MPEG_PS_PAL profile, then the UPnP AV MediaServer may provide a dlna:res@ifoFileURI attribute to present the URI for the IFO file as defined in 7.7.18. The namespace for DLNA defined properties must be “urn:schemas-dlna-org:metadata-1-0/” and the namespace prefix must be “dlna:”.	[31]	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.
	M	A	DMS	7.3.38.2 If MPEG_PS_NTSC or MPEG_PS_PAL profiled content contains discontinuous SCR and/or PTS, then the associated <res> element must provide a URI for the IFO file through the dlna:res@ifoFileURI attribute.	[31]	3rd party DLNA MPEG_PS_NTSC/PAL content (comes from another DMS) must be SCR/PTS-continuous if IFO file is not exposed with that 3rd party content.

Table 7-3 Media Management Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.3.39 M	M	A	DMS	7.3.39.1 If a UPnP AV MediaServer provides an IFO file, then the URI of the IFO file must be specified. IFO file URIs are governed by guidelines in 7.3.26, "MM URI Rules"). 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/MPEG/IFO_101.ifo	[31]	Rendering endpoints 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 MEDIA FORMAT PROFILES

Introduction

The content items of various DLNA Home Networked Device Interoperability Guidelines media classes (i.e., image, audio, AV) are formatted according to a well-defined set of DLNA media format profiles. These profiles and their associated media compression formats are listed in 7-5, "Media Format Profiles". Note that MIME types are always case-insensitive. Several DLNA media format profiles can be defined for a single media class. In addition, several DLNA media format profiles can be associated with a given media compression format (e.g., JPEG).

Chart 7-5 Media Format Profiles

Name	Description	Media Class	Media Format	MIME Type (Case-insensitive)	Comments
LPCM	Baseline profile for audio media class content	Audio	LPCM	audio/L16	Valid MIME-type syntax is fully described in guideline 7.5.1.1.
JPEG_SM	Baseline mandatory profile for image media class content	Image	JPEG	image/jpeg	
JPEG_MED	Profile for image media class content of medium resolution	Image	JPEG	image/jpeg	
JPEG_LRG	Profile for image media class content of high resolution	Image	JPEG	image/jpeg	
JPEG_TN	Profile for image thumbnails	Image (Icon)	JPEG	image/jpeg	
PNG_TN	Profile for image thumbnails	Image (Icon)	PNG	image/png	
JPEG_SM_ICO	Profile for small icons	Image (Icon)	JPEG	image/jpeg	
JPEG_LRG_ICO	Profile for large icons	Image (Icon)	JPEG	image/jpeg	
PNG_SM_ICO	Profile for small icons	Image	PNG	image/png	
PNG_LRG_ICO	Profile for large icons	Image	PNG	image/png	
MPEG_PS_NTSC	Baseline mandatory profile for NTSC-formatted AV class media	AV	MPEG-2	video/mpeg	
MPEG_PS_PAL	Baseline mandatory profile for PAL-formatted AV class media	AV	MPEG-2	video/mpeg	
MPEG_TS_SD_NA	North America region profile for Standard Definition AV class media with zero TTS as per 7.7.26	AV	MPEG-2	video/mpeg	
MPEG_TS_SD_NA_T	North America region profile for Standard Definition AV class media with a valid TTS as per 7.7.26	AV	MPEG-2	video/mpeg	

Chart 7-5 Media Format Profiles (Continued)

Name	Description	Media Class	Media Format	MIME Type (Case-insensitive)	Comments
MPEG_TS_HD_NA	North America region profile for High Definition AV class media with a zero TTS as per 7.7.26	AV	MPEG-2	video/mpeg	
MPEG_TS_HD_NA_T	North America region profile for High Definition AV class media with a valid TTS as per 7.7.26	AV	MPEG-2	video/mpeg	
MPEG_TS_SD_EU	European region profile for Standard Definition AV class media with a zero TTS as per 7.7.26	AV	MPEG-2	video/mpeg	
MPEG_TS_SD_EU_T	European region profile for Standard Definition AV class media with a valid TTS as per 7.7.26	AV	MPEG-2	video/mpeg	
MPEG_TS_SD_KO	Korea region profile for Standard Definition AV class media with a zero TTS as per 7.7.26	AV	MPEG-2	video/mpeg	
MPEG_TS_SD_KO_T	Korea region profile for Standard Definition AV class media with a valid TTS as per 7.7.26	AV	MPEG-2	video/mpeg	
MPEG_TS_HD_KO	Korea region profile for High Definition AV class media with a zero TTS as per 7.7.26	AV	MPEG-2	video/mpeg	
MPEG_TS_HD_KO_T	Korea region profile for High Definition AV class media with a valid TTS as per 7.7.26	AV	MPEG-2	video/mpeg	

MEDIA FORMAT INTEROPERABILITY MODEL

In order to provide a media interoperability model between various types of DLNA devices, DLNA media format profiles are classified into ***mandatory*** (or baseline) media format profiles and ***optional*** media profiles. 7-4, "Media Interoperability Guidelines" summarizes the guidelines for mandatory and optional profile usage by various types of DLNA devices and their associated media content.

A rendering endpoint that supports a given media class must support all mandatory media format profiling for that media class. However, such device may implement additional partial profiling of another media class. Such partial profiling can include just a subset of mandatory format profiles or just a subset of profile parameter sets. Such device would not be able to claim support of that additional media class. For example, a DLNA rendering endpoint that supports an audio media class must implement all mandatory audio media class format profiling (i.e., LPCM), but could also provide support for rendering video profiled by an optional AV media class format profile (MPEG_TS_SD_EU - profiled video).

Table 7-4 Media Interoperability Guidelines

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.4.1 MF Mandatory Media Formats: Support Requirements	M	C	DMP	7.4.1.1 Rendering endpoints that claim to support a particular DLNA media class must support all mandatory DLNA media format profiles for that media class.	[66]	A rendering endpoint designed to support only the DLNA audio media class is not required to support mandatory DLNA AV media format profiles.
	M	C	DMS	7.4.1.2 Serving endpoints that claim to support a particular DLNA media class must support at least one of the mandatory DLNA media format profiles for that media class.	[66]	A serving endpoint that cannot expose and stream some content in a mandatory DLNA media format profile is not compliant with this requirement. A compliant implementation must have some means to allow content to be exposed and streamed in at least one of the mandatory DLNA media format profiles for the DLNA media classes it claims to support. Examples of compliant serving endpoint implementations include: • A serving endpoint implementation on a PC that allows the user or application acting on the user's behalf to provide content in at least one of the mandatory DLNA media format profiles for the media classes the serving endpoint implementation claims to support so that it can be exposed and streamed by the serving endpoint implementation to other DLNA devices.

Table 7-4 Media Interoperability Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
						• A serving endpoint device that allows the encoding of content to be user selectable in at least one of the mandatory DLNA media format profiles for the DLNA media classes it claims to support. A Digital Video Recorder might by default support MPEG-4 encoding of captured video content , but either convert to a mandatory DLNA AV media format profile for on-demand streaming to DLNA devices or else allow the user to change the content capture encoding to a mandatory DLNA AV media format profile. Such a device would also be compliant if the user can insert a memory card into it with content in a mandatory DLNA AV media format profile and the device then is able to expose and steam it to other DLNA devices.
7.4.2 MF Optional Media Formats: Support Requirements	O	C	DMS,DMP	7.4.2.1 Rendering endpoints and serving endpoints that claim to support a particular DLNA media class may support any of the optional DLNA media format profiles for that media class.	[66]	

Table 7-4 Media Interoperability Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.4.3 MF Optional Media Formats: Content Availability	S	C	DMS	7.4.3.1 If a serving endpoint exposes a content item encoded to an optional DLNA media format profile in a CDS with the DLNA.ORG_PN parameter in res@protocolInfo, then the serving endpoint should also expose this content in a mandatory DLNA media format profile for the relevant media class. It is recommended that all content available on serving endpoints be exposed if possible in a mandatory DLNA media format profile regardless of its native format.	[66]	Since rendering endpoints such as a DMP are required to support content in mandatory DLNA media format profiles, It is recommended that all content available in optional DLNA media format profiles on serving endpoints also be made available and exposed in a mandatory DLNA media format profile. This may be achieved by a variety of means including storing copies of a content item in different formats or by conversion on demand for a content item.

Table 7-4 Media Interoperability Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.4.4 MF Optional Media Formats: User Indications	M	C	DMP	7.4.4.1 For any content item in an optional DLNA media format profile, an indication must be given to the user of of the following condition at the time content items are exposed to the user for selection: • If the selected rendering endpoint is not capable of rendering the content item. The form of this user indication is implementation dependent and can be user selectable.	[66]	Examples of implementations that comply with this requirement include: • Not displaying a content item that meets the condition to the user. • Graying out a content item that meets the condition and preventing its selection for playback by the user. Note that this requirement works for a serving endpoint controller (i.e. a MSCP) that is bound to a rendering endpoint (e.g. a DMP) but in the future may have to be revised, or another guideline added, to cover other control point configurations. User selectable means that an implementation may allow users to turn this indication feature on or off, or an implementation may allow users to select among multiple indication alternatives. The actual methods for providing these options to users and the default values for these options are implementation dependent.

Table 7-4 Media Interoperability Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
	M	C	DMP DMS	7.4.4.2 For any content item in an optional DLNA media format profile, an indication must be given to the user of of the following condition at the time content items are exposed to the user for selection: • If the content item is not available in a mandatory DLNA media format profile. The form of this user indication is implementation dependent and can be user selectable.	[66]	Examples of implementations that comply with this requirement include: • Not displaying a content item that meets the condition to the user. • A textual statement for the user indicating a content item that meets the condition is not available in a mandatory DLNA format profile. • A visual mark for the user such as a '-' or '*' displayed alongside a content item that meets the condition and explanatory text elsewhere (e.g. a product manual or help file) Note that this requirement works for a serving endpoint controller (i.e. MSCP) that is bound to a rendering endpoint (e.g, a DMP), or if present, a serving endpoint controller application bound to the serving endpoint (e.g.. DMS). In the future this guideline may have to be revised, or another guideline added, to cover other control point configurations.User selectable means that an implementation may allow users to turn this indication feature on or off, or an implementation may allow users to select among multiple indication alternatives. The actual methods for providing these options to users and the default values for these options are implementation dependent.

7.5 AUDIO MEDIA CLASS RELATED MEDIA FORMAT PROFILES

The content of the DLNA Home Networked Device Interoperability Guidelines Audio media class must be formatted according to the mandatory baseline audio media format guideline specification of 7-5, "Mandatory Audio Media Class Profile Guidelines".

Various combinations of media format compression-space and file / system-level parameters and attributes form a so-called "Profile Parameter Sets". Profile Parameter Sets are defined by a specific combination of such parameters or attributes and are indicated by "➢" symbol throughout the media format sections of the Interoperability Guidelines.

As the general rule that must be followed for Audio media class interoperability, rendering endpoints (e.g., DMP) must support rendering of all mandatory profile parameter sets in a given audio media class profile. Serving endpoints (e.g., DMS) must expose their DLNA content formatted according to at least one profile parameter set in a given audio media class profile. For example, two profile parameter sets that specify the LPCM audio sample rates for the DMP device class are defined in the entries below: 44.1 kHz and 48 kHz. This means that DMP must provide rendering capabilities for both LPCM sample rates, while DMS is free to make its audio media content available in either one of the sample rates (e.g., 48 kHz only).

In certain instances, simple tolerance (i.e., acceptance) rather than rendering of certain profile parameter sets is allowed for rendering endpoints. Such cases are indicated specifically within the guideline entries of 7-5.

Table 7-5 Mandatory Audio Media Class Profile Guidelines

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.5.1 MF Baseline LPCM Audio Format: DLNA Profile **LPCM**	M	L	DMS, DMP	7.5.1.1 This baseline audio format specification must follow profiling of **[35]**, which defines the MIME encapsulation for the LPCM DLNA media format and **[48]** uses the "L16" audio media format defined by **[48]**. "L16" denotes uncompressed audio data, using 16-bit signed representation in two's-complement notation and network byte order. There are the following parameter constraints to "L16" as defined by DLNA. **Sampling Rates:** ➢ c44.1 kHz ➢ 48 kHz **Number of channels:** ➢ 1 (if the content is mono) ➢ 2 (if the content is stereo) **Quantization:** 16 bit MIME type "audio/L16" must be used for this media format. The "channels" parameter should be included in MIME type header exposed by a serving endpoint. If a serving endpoint does not include the "channels" parameter in a content description, the default value assumed by the rendering endpoint must be 1. The "rate" parameter must be included in MIME type header exposed by the serving endpoint. **[35]** also defines some additional parameters to audio/L16, (i.e. emphasis, channel-order, etc). These parameters must not be used as MIME type parameters (i.e., they are illegal in DLNA). Example MIME type for stereo LPCM content of 44.1 kHz sampling rate is: audio/L16;rate=44100;channels=2 DLNA MF profileID "LPCM" must be used to indicate the content of this DLNA media format profile.	**[35]** **[48]**	Note that MIME types are case-insensitive. Therefore, "audio/L16" or "audio/l16" are both acceptable MIME type usages in this requirement. Sample rate and the number of channels parameters are provided in the MIME type header. The "channels" parameter is not required by **[35]**. The default channel ordering for 2 channel content is: Channel 1: Left Channel 2: Right as indicated by**[35]**.

Table 7-5 Mandatory Audio Media Class Profile Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.5.2 MF Baseline LPCM Audio Format DLNA Profile **LPCM** HTTP encapsulation requirement	M	C	DMS	7.5.2.1 The first byte of an HTTP response carrying LPCM content in its entity body to a client must be aligned to the samples' boundary (for monaural stream) or stereo set of samples (for stereo stream), unless **RANGE** header field is used in the HTTP request and the specified starting index is not aligned to the boundary.		

7.6 IMAGE MEDIA CLASS RELATED MEDIA FORMAT PROFILES

The content of the DLNA Home Networked Device Interoperability Guidelines Image media class must be formatted according to the baseline image media format profile specified in 5-1, "DLNA Device Classes" and could be optionally formatted according to the optional image media format profile specification shown in 5-1.

Various combinations of media format compression-space and file/system-level parameters and attributes form a so-called "Profile Parameter Sets". Profile parameter sets are defined by a specific combination of such parameters or attributes and are indicated by "➢" symbol throughout the media format section of this guideline.

As the general rule that must be followed for image media class content interoperability, rendering endpoints (e.g., DMP) must support rendering of all mandatory profile parameter sets for a given DLNA image media profile. Serving endpoints (e.g., DMS) must expose their DLNA content formatted according to at least one profile parameter set of the mandatory image media class profile formatting.

Thus, at a minimum, rendering endpoints must support rendering of the image media class according to all forms (i.e., profile parameter sets) of the DLNA JPEG_SM media profile as indicated by table 5-1. Chart 7-6 summarizes the content availability requirements for the image class. Serving endpoints must expose image content in at least one profile

parameter set of the DLNA JPEG_SM media format. Serving endpoints may also provide image resource of the same content compliant to DLNA JPEG_MED, DLNA JPEG_LRG, DLNA JPEG_TN, or DLNA PNG_TN image profiles. Rendering endpoints may optionally have the capability to render DLNA JPEG_MED, DLNA JPEG_LRG, DLNA JPEG_TN, or DLNA PNG_TN formatted images.

DLNA icons must be formatted according to PNG_SM_ICO, PNG_LRG_ICO, JPEG_SM_ICO and JPEG_LRG_ICO image media format profiles. Different rules apply toward icon formatting. If icons are supported by DLNA endpoints, icons must be made available in all four DLNA icon media format profiles. For detailed guideline specification, see 7.7.27 "DDC UPnP Device Icons".

Chart 7-6 Image Media Class Media Format Profile Content Availability

		Original Content to be Exposed as DLNA Content is in this DLNA Media Format Profile		
		JPEG_SM	JPEG_MED	JPEG_LRG
UPnP AV MediaServer Exposed Content Media Format Profile	JPEG_SM	✓	✓	✓
	JPEG_MED		✓	
	JPEG_LRG			✓

✓ Content Availability in DMS

Table 7-6 Image Media Class Guidelines

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
General Capability Requirements						
7.6.1 MF Baseline JPEG_SM Image Format Profile: Profile **JPEG_SM**	M	L	DMS, DMP	7.6.1.1 The DLNA JPEG_SM must follow the following requirements. **File Format:** ➢ EXIF Ver.1.x or later ➢ JFIF 1.02 **Image compression:** The primary image data compression (even in JFIF or EXIF 1.x file format) must be conformant to EXIF Ver.2.21 [37], [38] with the following additional constraints; **Maximum Resolution:** The resolution of the image must not exceed 640x480 (i.e., horizontal pixels * vertical pixels) **Huffman table:** The typical Huffman table defined by JPEG standard **Color space:** ➢ sRGB ➢ Uncalibrated color space. MIME type “image/jpeg” must be used for this media format. DLNA MF profileID “JPEG_SM” must be used to indicate the content of this media format profile.	[36] [37] [38] [39] [40]	The constraints to JPEG compression is equivalent to those of DCF. Digital still camera creates image compliant to DCF [40]. The max resolution, 640x480 is selected for a resource-constrained rendering endpoint. Dedicated photo players (e.g. HDTV photo player) should decode higher-resolution images if a DMS provides DLNA JPEG_MED or DLNA JPEG_LRG – formatted images in addition to DLNA JPEG-640. Serving endpoint is allowed to expose images file-formatted according to earlier versions of EXIF (i.e., EXIF 1.x), due to backward compatibility of those file formats. The compression aspects of such images are not covered by this relaxation rule.

Table 7-6 Image Media Class Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.6.2 MF Optional Image Formal Profiles: Profiles: **JPEG_MED**, **JPEG_LRG**, **JPEG_TN**, **PNG_TN**, **JPEG_SM_ICO**, **JPEG_LRG_ICO**, **PNG_SM_ICO**, and **PNG_LRG_ICO**	0		DMS DMP	7.6.2.1 The contents of an Image media class may be profiled according to the optional DLNA media format profiles, **JPEG_MED**, **JPEG_LRG**, **JPEG_TN**, **PNG_TN**, **JPEG_SM_ICO**, **JPEG_LRG_ICO**, **PNG_SM_ICO**, **and PNG_LRG_ICO,** as indicated in the guideline entries 7.6.3 through 7.6.10.		
7.6.3 MF Optional JPEG_MED Image Format Profile: Profile **JPEG_MED**	M	L	DMS, DMP	7.6.3.1 The DLNA JPEG_MED media format must follow the requirement of 7.6.1 DLNA JPEG_SM baseline format except for the following image resolution requirements. **Maximum Resolution:** Number of pixels must not exceed 1024*768 (horizontal pixels * vertical pixels) MIME type “image/jpeg” must be used to indicate the content of this media format profile. DLNA MF profileID “JPEG_MED” must be used to indicate the content of this media format profile	[36] [37] [38] [39] [40]	
7.6.4 MF Optional JPEG_LRG Image Format Profile: Profile **JPEG_LRG**	M	L	DMS, DMP	7.6.4.1 The DLNA JPEG_LRG media format must follow the requirement of 7.6.1 DLNA JPEG_SM baseline format except for the following image resolution requirements. **Maximum Resolution:** Number of pixels must be up to 4096*4096 (horizontal pixels * vertical pixels) MIME type “image/jpeg” must be used for this media format, DLNA MF profileID “JPEG_LRG” must be used to indicate the content of this media format profile.	[36] [37] [38] [39] [40]	

Table 7-6 Image Media Class Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.6.5 MF Optional JPEG_TN Image Format Profile: Profile **JPEG_TN**	M	L	DMS, DMP	7.6.5.1 The DLNA JPEG_TN media format must follow the requirements of 7.6.1 DLNA JPEG_SM baseline format except for the following requirements. In this JPEG_TN format file, the thumbnail image must be stored as the *primary* image data of the file. **Maximum Resolution:** The resolution of the image must not exceed 160*160 (i.e., horizontal pixels * vertical pixels). MIME type “image/jpeg” must be used for this media format. DLNA MF profileID “JPEG_TN” must be used to indicate the content of this media format profile. The DLNA JPEG_TN media format must be used only for a thumbnail representation of some content.	[36] [37] [38] [39] [40]	
7.6.6 MF Optional PNG_TN Image Format Profile: Profile **PNG_TN**	M	L	DMS, DMP	7.6.6.1 The PNG_TN media format profile must follow the requirements of 7.6.9 PNG_SM_ICO media format profile specification, except for the following requirements. **Maximum Resolution:** The resolution of the image must not exceed 160*160 (i.e., horizontal pixels * vertical pixels). MIME type “image/png” must be used for this media format. DLNA MF profileID “PNG_TN” must be used to identify content of this profile. The PNG_TN media format profile must be used only for a thumbnail representation of some content.	[49] [50]	
7.6.7 MF Optional JPEG_SM_ICO Image Format Profile: Profile **JPEG_SM_ICO**	M	L	DMS, DMP	7.6.7.1 The JPEG_SM_ICO media format profile must follow the requirements of 7.6.1 DLNA JPEG_SM baseline media format profile, except for the following requirements. **Maximum Resolution:** The resolution of the image must be 48*48 (i.e., horizontal pixels * vertical pixels). It should also be noted that this media format profile is for use with the 7.2.27 DDC UPnP Device Icons guideline and other guidelines in the future that call for an icon image media format profile. ProfileID is not used.	[36] [37] [38] [39] [40]	This media format profile is essentially the same as the JPEG_SM profile, except it specifies a very small image size for use in scenarios involving small icons.

Table 7-6 Image Media Class Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.6.8 MF Optional JPEG_LRG_ICO Image Formal Profile: Profile **JPEG_LRG_ICO**	M	L	DMS, DMP	7.6.8.1 The JPEG_LRG_ICO media format profile must follow the requirements of 7.6.1 DLNA JPEG_SM baseline media format profile, except for the following requirements. **Maximum Resolution:** The resolution of the image must be 120*120 (i.e., horizontal pixels * vertical pixels). It should also be noted that this media format profile is for use with the 7.2.27 DDC UPnP Device Icons guideline and other guidelines in the future that call for an icon image media format profile. ProfileID is not used.	[36] [37] [38] [39] [40]	This media format profile is essentially the same as the JPEG_SM profile, except it specifies a smaller image size for use in scenarios involving large icons.

Table 7-6 Image Media Class Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.6.9 MF Optional PNG_SM_ICO Image Format profile: Profile **PNG_SM_ICO**	M	L	DMS, DMP	7.6.9.1 The PNG_SM_ICO media format profile must follow the requirements of **[50]** recommendations as well as the following restrictions and guidelines described in the entries below. **Maximum Resolution:** The resolution of the image must be 48*48 (i.e., horizontal pixels * vertical pixels). The MIME type for this image profile must be "image/png". The image must use one of the following color types, defined in section 6.1 of **[49]**. ➢Greyscale (with or without tRNS chunks) ➢Truecolor (with or without tRNS chunks) ➢Indexed-color (with or without tRNS chunks) ➢Greyscale with alpha ➢Truecolor with alpha The total number of bits (excluding alpha channel bits) needed to represent a color must not exceed 24 bits, as described below. ➢Greyscale: 8 or 16 bits ➢Truecolor: 24 bits (triplet of 8 bit R/G/B samples) ➢Indexed-color: 24 bits (palette entry is a triplet 8 bit R/G/B samples) ➢Greyscale with alpha: 8 or 16 bits (with matching alpha channel depth) ➢Truecolor with alpha: 24 bits (triplet of 8 bit R/G/B samples, alpha channel must be 8 bits) The image must use "interlace method 0" (also known as the null method), as described in section 8.2 of reference **[49]**. It should also be noted that this media format profile is for use with the 7.2.27 DDC UPnP Device Icons guideline and other guidelines in the future that call for an icon image media format profile. ProfileID is not used.	**[49]** **[50]**	The W3C Recommendation of the PNG specification **[50]** is also International Standard, ISO/IEC 15948:2003 **[49]**. The purpose of this media format profile is to define a subset of the PNG specification for UPnP device icons and image thumbnails. PNG's capability for lossless compression and transparency makes it ideal for extremely small image sizes. Rendering endpoints for this media profile does not need to claim "absolute requirements" for PNG renderer as per section 13.1 of **[50]**. PNG decoder implementations can be built specifically for this media format profile and not claim full compliance to the PNG specification. This guideline requirement allows for transparency. This guideline requirement specifies the bit-depth individual sampling channels as well as the total bit depth for a color. Since the home network generally has higher bandwidth and lower latency than the Internet, "interlace method 1" (also known as Adam7) is less useful.

Table 7-6 Image Media Class Guidelines (Continued)

Name	M/S/O	A/C/F/L/R	Dev Class	Description	Ref #	Comments
7.6.10 MF Optional PNG_LRG_ICO Image Format Profile: Profile **PNG_LRG_ICO**	M	L	DMS, DMP	7.6.10.1 The PNG_LRG_ICO media format profile must follow the requirements of 7.6.9 PNG_SM_ICO media format profile, except for the following requirements. **Maximum Resolution:** The resolution of the image must be 120*120 (i.e., horizontal pixels * vertical pixels). It should also be noted that this media format profile is for use with the 7.2.27 DDC UPnP Device Icons guideline and other guidelines in the future that call for an icon image media format profile. ProfileID is not used.	[49] [50]	

7.7 AV MEDIA CLASS RELATED MEDIA FORMAT PROFILES

The DLNA Home Networked Device Interoperability Guidelines compliant content of the AV media class must be formatted according to the mandatory baseline AV media format profiles specified in 7-5 and could be optionally formatted according to the optional AV media format profile specification 7-6.

Various combinations of AV media format compression-space and system-level parameters and attributes form a so-called “Profile Parameter Sets”. Profile parameter sets are defined by a specific combination of such parameters or attributes and are indicated by “➢” symbol throughout the media format sections of the Interoperability Guidelines. For example, different combinations of audio sampling rates and video resolutions / aspect ratios form multiple different profile parameter sets.

As the general rule that must be followed for AV content interoperability between DLNA devices, if the media profile is supported by the rendering endpoint, rendering endpoints must support rendering of all mandatory profile parameter sets of this media profile. Serving endpoints (e.g.,

DMS) can expose their content formatted according to at least one profile parameter set of a given AV media class profile.

AV media class profile consists of the video portion and the audio portion profiling. Different digital encoding formats of the audio portion for AV media class content may exist. As an example, “LPCM”, “MPEG-1/2” and “AC-3” are all different audio portion profiling options for the MPEG-2 Program Stream (PS)-profiled AV media. Depending on the DLNA media format profile in question, audio portion profiling may be mandatory or optional for the rendering endpoint (e.g. DMP). For example, “LPCM”, “MPEG-1/2” and “AC-3” are all mandatory for rendering endpoint in case of MPEG_PS_NTSC and MPEG_PS_PAL DLNA profiles. However, “AC-3” support is optional by the rendering endpoint in case of MPEG_TS_SD_EU DLNA format profile. Various guideline entries of 7-5 and 7-6 reflect these requirements.

In certain instances, simple tolerance (i.e., acceptance) rather than rendering of certain profile parameter sets is allowed for a rendering endpoint. Such cases are indicated specifically within the guideline entries of 7-5 and 7-6.

Two different types of DMPs, capable of rendering AV content, are defined for different geographical regions. At a minimum, the first type, DMP_A, supports the mandatory baseline AV media class profiling according to 7-5. The second type, DMP_B, also includes support for the AV media class profiling according to 7-6 according to the relevant geographical region. 7-5 illustrates which media format profiles are supported by a DMS, DMP_A, and DMP_B per different geographical regions (namely, Japan, United States, Korea, and Europe). As seen from this table DMP_A is not a valid configuration in the Korea region.

A number of the guidelines in 7-7 refer to the TV systems: PAL and NTSC. These guidelines do not refer to the analog part of the respective TV system, but only to the TV system-related relevant digital encoding characteristics, like the following parameters:

(1) “TV System PAL” in DLNA context refers to 25 Hz video frame rate. (2) “TV System NTSC” refers to 29.97 Hz video frame rate.

Chart 7-7 Media Format Profiles Supported by DMS and DMPs by Region

		Japan	US	Korea	EU
DMS		**Conversion of Transport Stream (TS) contents to Program Stream (PS) is not required.**			
DMP_A	**PS**	**Required (MPEG_PS_NTSC)**	**Required (MPEG_PS_NTSC)**	**Not Applicable (NA)**	**Required (MPEG_PS_PAL)**
	TS	**Optional**	**Optional**	**Not Applicable (NA)**	**Optional**
DMP_B	**PS**	**Required (MPEG_PS_NTSC)**	**Required (MPEG_PS_NTSC)**	**Required (MPEG_PS_NTSC)**	**Required (MPEG_PS_PAL)**
	TS	**Required (Future Interoperability Guidelines)**	**Required (MPEG_TS_SD_NA or MPEG_TS_SD_NA_T)**	**Required (MPEG_TS_SD_KO or MPEG_TS_SD_KO_T)**	**Required (MPEG_TS_SD_EU, MPEG_TS_SD_EU_T)**

[

Table 7-7 AV Media Class Formatting Guidelines

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
Mandatory General Format System-level Guidelines						
7.7.1 MF Baseline AV MPEG-2 Format Profile: System Stream	M	L	DMP, DMS	7.7.1.1 MPEG-2 Program Stream (PS) must be in accordance with the constraints as described in this Table.	[41] [42] [43] [44] [45] [47]	
7.7.2 MF Baseline AV MPEG-2 Format Profile: Program Stream Profiling	M	L	DMS	7.7.2.1 The Program Streams (PS) must be profiled according to the video format-specific guideline requirements 7.7.3 through 7.7.14 and at least one of the following audio profile parameters: LPCM 7.7.15, AC-3 7.7.16, and MPEG Audio Layer 2 7.7.17.	[41] [42] [43] [44] [45]	
	M	L	DMP	7.7.2.2 Rendering endpoints must support (render) the MPEG-2 Program Stream (PS) profiled according to all video format-specific profile parameter sets listed in 7.7.3 through 7.7.14 and all audio portion profiles (i.e., LPCM, MPEG-1/2-Layer 2, and AC-3) specified in the entries 7.7.15 through 7.7.17. The following are the exception rules: **TV system (NTSC / PAL)** Rendering endpoints may support either NTSC or PAL. **AC-3** Rendering endpoints must accept all Content Audio Channel Modes, however only 2 Output Audio Channel modes are mandatory for a rendering endpoint, as will be described in following guideline entries. **MPEG-2 L2 extension stream** Rendering endpoints must accept extension stream but not necessarily render multichannel information in it.	[41] [42] [43] [44] [45]	

Table 7-7 AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.3 MF Baseline AV MPEG-2 Format Profile: Program Bit Stream Profiling	M	R	DMS, DMP	7.7.3.1 The PS bit stream must consist of a sequence of MPEG-2 Packs, as defined in **[41]**, each of which may carry audio, video, or other data.	**[41]**	
7.7.4 MF Baseline AV MPEG-2 Format Profile: Program Bit Stream Structure: Stream and Substream IDs	M	C	DMS, DMP	7.7.4.1 Serving endpoints and rendering endpoints must rely on the stream_id and sub_stream_id field to identify video and audio elementary streams as defined in this document according to **[47]**, and not in the Program Stream Map (PSM) or Directory structures of **[41]**.	**[41]** **[47]**	
7.7.5 MF Baseline AV MPEG-2 Format Profile: Program Bit Stream Structure: Data Packs	M	C	DMS, DMP	7.7.5.1 Serving endpoints may insert RDI and sub-picture packs in agreement to the provisions defined in **[47]**. Rendering endpoints must tolerate these additional data-only packs. In this case “tolerate” means “to discard gracefully” if the rendering endpoint is unable to decode or interpret the packs.	**[47]**	This entry makes navigation and subpicture packs optional. They will not be generated for example in cases of TS to PS conversions
7.7.6 MF Baseline AV MPEG-2 Format Profile: Program Bit Stream Structure: Private Packs	M	C	DMS, DMP	7.7.6.1 Serving endpoint may also insert additional private packs recognized by the use of private stream_id and/or substream_id values. Rendering endpoints must tolerate additional private packs; discarding packs gracefully if unable to decode or interpret. Serving endpoints that insert private packs should use the stream_id extensions defined in **[46]** to avoid potential conflicts with future usage of stream_id and substream_id fields.	**[46]** **[47]**	This entry clarifies that proprietary information may exist in the stream. It recommends the preferred MPEG-2 method to add private information without overloading stream_id values.
7.7.7 MF Baseline AV MPEG-2 Format Profile: Program Stream Encapsulation	M	C	DMS	7.7.7.1 The first byte of an HTTP response carrying media content of this profile in its entity body to a client must be aligned to the MPEG-2 pack boundary, unless **RANGE** header field is used in the HTTP request, and the specified starting index in it is not aligned to the pack boundary.	**[41]** **[47]** **[19]**	This entry clarifies the HTTP-based encapsulation of MPEG-2 Program Streams.

Table 7-7 AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.8 MF Baseline AV MPEG-2 Format Profile HTTP Alignment for Time Seek operations	S	L	DMS	7.7.8.1 When TimeSeekRange.dlna.org header field is used in the HTTP request, the first byte of an HTTP response carrying media content of this profile in its entity body to a client should be aligned to the GOP boundary.		
7.7.9 MF Baseline AV MPEG-2 Format Program Stream: Best Effort Source Streams	M	C	DMS	7.7.9.1 Elementary Streams at the serving endpoint prior to transmission must conform to the constraints imposed by the Buffer Reference Models defined in **[47]**.	**[47]**	The network may introduce jitter due to best effort traffic characteristics. This clarification explains that serving endpoints can stream the program multiplex using best effort conditions, and may stream content at higher or lower rates than the consumption rate.
7.7.10 MF Baseline AV MPEG-2 Format Profile: Program Stream: Best Effort Destination Streams	M	C	DMP	7.7.10.1 On arrival, rendering endpoints may reconstruct the original program multiplex typically using buffers that compensate for network jitter, delays, etc. The rate for the program multiplex must be obtained from the program_mux_rate field of a pack header. The reconstruction process is device implementation-specific,	**[47]**	This statement says that rendering endpoints will rely on the information from one particular field to reconstruct the program mux before decoding.
7.7.11 MF Baseline AV MPEG-2 Format Profile: MIME Type Definition	M	A	DMS, DMP	7.7.11.1 MIME type “video/mpeg” must be used for this media format.		
7.7.12 MF Baseline AV MPEG-2 Format Profile: Profiles **MPEG_PS_NTSC** and **MPEG_PS_PAL**	M	A	DMS, DMP	7.7.12.1 The following dlna-profile must be used for this media format. If the TV System is NTSC: DLNA profile=MPEG_PS_NTSC If the TV System is PAL: DLNA profile=MPEG_PS_PAL		If the TV system supports both NTSC and PAL, it can be expose using both MPEG_PS_NTSC and MPEG_PS_PAL DLNA profiles.

Table 7-7 AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
Mandatory Format Compression-level Requirements						
7.7.13 MF Baseline AV MPEG-2 Format Profile: System Stream Specification	M	A	DMS, DMP	7.7.13.1 Main characteristics of MPEG-2 system stream are: **System** Program Stream **Number of Video Streams** 1 video stream only **Number of Audio Stream** Up to 2 audio streams **System bit rate** Up to 10.08 Mbps **Pack Size** Must be 2048 bytes per each Pack **Pack Header** Must match the provisions defined “ in Table 5.3.1-2: Pack Header in [47] Audio Pack Audio Pack Structure must match the provisions in Figure 5.3.3-1 : Structure of Audio Pack in [47] **Video Pack** Video Pack Structure must match the provisions in Figure 5.3.2-1 : Structure of Video Pack in [47]. **System Header** Must match the provisions defined in Table 5.3.2-1: System header in [47] . **Program End Code** Program end code may be included at the end of stream as in [47] . **Content length** Must not exceed 1/90000 x 4294967295 (2^32-1) second if the content has no corresponding IFO file.	[41] [47]	Stuffing up to 2048 bytes pack size must be done at the packet header or with padding packet. If the content has a corresponding IFO file, the content length limitation corresponds to that in the IFO file

Table 7-7 AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.14 MF Baseline AV MPEG-2 Format Profile: Video Elementary Stream Specification Profiles **MPEG_PS_NTSC** and **MPEG_PS_PAL**	M	A	DMS, DMP	7.7.14.1 Video Elementary Stream must match the provisions about MPEG-2 Video Stream in sections 1.2 , 5.1 , 5.2.1 , 5.3.2 and 5.5.1.2 of [47]. Main characteristics of MPEG-2 video stream are: **Profile:** MP@ML **Stream ID in MPEG-2 System** 0xE0. (defined in Table 5.1-3 of [47]: stream_id). **Sequence_header** Sequence_header followed by GOP_header must be set at the beginning of video data in video pack at the intervals between 0.4 seconds and 1.0 seconds.(defined in 5.2.1 Movie VOB in [47]) **Video bit rate** CBR: less than or equal to 9.80 Mbps **VBR:** maximum bit rate less than or equal to 9.80 Mbps (defined in Table 5.5.1.2-1 : Constraints on MPEG-2 video in [47]) **TV System:** NTSC or PAL with following characteristics.	[41] [42] [47]	Serving endpoints and DMP may support either NTSC or PAL e.g., PTS[32] means the specific bit (i.e., MSB in this case) of the PTS field.

	NTSC	PAL
RESOLUTION	➢ 720*480 ➢ 704*480 ➢ 544*480 ➢ 480*480 ➢ 352*480 ➢ 352*240	➢ 720*576 ➢ 704*576 ➢ 544*576 ➢ 480*576 ➢ 352*576 ➢ 352*288
	THE RESOLUTION MAY BE CHANGED DURING A SINGLE CONTENT STREAMING SESSION. WHEN THIS HAPPENS, AND THERE IS NO ASSOCIATED IFO FILE, THEN THE SEQUENCE_END_CODE MUST BE SET TO THE BOUNDARY OF RESOLUTION CHANGE.	
FRAME RATE	➢ 29.97HZ	➢ 25HZ
NUMBER OF PICTURES IN GOP	➢ 36 DISPLAY FIELDS OR LESS 18 FRAMES OR LESS (PROGRESSIVE)	➢ 30 DISPLAY FIELDS OR LESS 15 FRAMES OR LESS (PROGRESSIVE)

Table 7-7 AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
				(These constraints are defined in Table 5.5.1.2-1 : Constraints on MPEG-2 Video in [47]) **Display Aspect Ratio** 4:3 16:9 Aspect ratio may be changed. When this happens, and there is no associated IFO file, then the sequence_end_code must be set to the boundary of aspect ratio change. (defined in 5.5.1.2 MPEG-2 video data for one GOP and Table C-2 : Relation between Presentation Data of Video and the Player in [47]) “low_delay” 0b i.e. “low_delay” sequences are not permitted. (defined in Table 5.5.1.2-1 : Constraints on MPEG-2 video in [47]) PTS, DTS in Video Packet PTS and DTS are mandatory in each Video Pack containing the first byte of the picture start code of any MPEG-2 I-picture encoded as a frame picture or any first MPEG-2 I-picture of a pair of two encoded field pictures. PTS[32] and DTS[32] must be set to zero.(defined in Table 5.3.2-2 : Video Packet in [47]) Video Packet Header Video Packet Header must match the provisions defined in Table 5.3.2-2 : Video packet in [47]		

Table 7-7 AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.15 MF Baseline AV MPEG-2 Format Profile: Audio Portion Profiling: **LPCM** Profiles **MPEG_PS_ NTSC** and **MPEG_PS_ PAL**	M	C	DMS, DMP	7.7.15.1 LPCM Audio Stream must match the provisions about LPCM Audio in 1.2 General Specifications of Presentation Data, 5.1 Definition of VOB, 5.3.3 Audio Pack and 5.5.2.1 Linear PCM audio in [47]. Main characteristics of LPCM audio stream are: **Scheme to combine the LPCM elementary stream to MPEG-2 Program Stream.** LPCM Audio bit stream is multiplexed as private_stream_1 Stream_id for main audio is 0xBD (indicating to the private_stream_1) Sub_stream_id to distinguish the LPCM audio stream is 0b1010_000*. (The first byte in data area of each packet is assigned as sub_stream_id). 0b1010_000* : * corresponds to audio stream number. *=0 for main language. *=1 corresponds to auxiliary language. (defined in 5.1 Definition of VOB in [47]) **Sampling Rate** 48 kHz (defined in Table 5.5.2.1-1 Linear PCM coding basics in [47].) **Audio Channel Mode** ➢ 2(stereo, dual monaural) ➢ 1(mono) (These are defined in Table 5.5.2.1-2: Linear PCM data size in a packet in [47].) **Quantization** 16 bits.(defined in Table 5.5.2.1-1 Linear PCM coding basics in [47].) **Audio bit rate** 1.536 Mbps per one stream (2 channels). 768 kbps per stream (mono)	[47]	

Table 7-7 AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
				Emphasis May be applied (zero point: 50 micro sec., Pole: 15 micro sec) (defined in Table 5.5.2.1-1 Linear PCM coding basics in [47].) **Channel assignment** In the stereo presentation mode, the descriptions of channels, ACH0 and ACH1 correspond to Left channel (L-ch) and Right channel (R-ch) respectively, (defined in F5.5.2.1 Linear PCM audio in [47]). **PTS in Audio Packet** PTS must be described in every Audio packet in which the first sample of audio frame is included. PTS[32] must be set to ZERO.(defined in Table 5.3.3-1 Audio Packet of Linear PCM in [47]) **Packet Header** Packet Header must match the provisions defined in Table 5.3.3-1: Audio packet of Linear PCM in [47]		

Table 7-7 AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.16 MF Baseline AV MPEG-2 Format Profile: Audio Portion Profiling: **AC-3** Profiles **MPEG_PS_ NTSC** and **MPEG_PS_ PAL**	M	C	DMS, DMP	7.7.16.1 AC-3 Audio Stream must match the provisions about AC-3 Audio in 1.2 General Specifications of Presentation Data, 5.1 Definition of VOB, 5.3.3 Audio Pack and 5.5.2.2 AC-3 audio in ?[47]. Main characteristics of AC-3 audio stream are: **Scheme to combine the AC-3 elementary stream to MPEG-2 Program Stream.** AC-3 Audio bit stream is multiplexed as private_stream_1 Stream_id for main audio is 0xBD (indicating the private_stream_1) Sub_stream_id to distinguish the AC-3 audio stream is 0b1000_000*. (The first byte in data area of each packet is assigned as sub_stream_id). 0b1000_000* : * corresponds to audio stream number. *=0 for main language. *=1 corresponds to auxiliary language. (defined in 5.1 Definition of VOB in [47]) **Sampling Rate** 48 kHz (defined in Table 5.5.2.2-1: Restricted Items for AC-3 of audio in [47]). **Content Audio Channel Modes** ➢ 1+1 ➢ 1/0 ➢ 2/0 ➢ 3/0 ➢ 2/1 ➢ 3/1 ➢ 2/2 ➢ 3/2 Changing audio channels among mono, dual-mono and stereo is allowed, (defined in Table 5.5.2.2-1: Restricted Items for AC-3 of audio and Table C-4 : Relation between Presentation Data of Audio and the Player in [47]).	[45] [47]	Content is converted from a Content Audio Channel Mode to an appropriate Output Audio Channel Mode by the AC-3 decoder.

Table 7-7 AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
				Output Audio Channel Modes ➢ 2 channels ➢ 1 channel **Audio bit rate:** 64 kbps - 448 kbps. (defined in Table 5.5.2.2-1 Restricted Items for AC-3 of audio in [47]) **Emphasis** Rendering endpoint may apply the emphasis if the emphasis flag is set. Encode device should encode audio contents without emphasis. **PTS in Audio Packet** PTS must be described in every audio packet in which the first sample of audio frame is included. PTS[32] must be set to zero. .(defined in Table 5.3.3-2 Audio Packet of AC-3 in [47]) **Packet Header** Packet Header must match the provisions defined in Table 5.3.3-2 : Audio packet of AC-3 in [47].		

Table 7-7 AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.17 MF Baseline AV MPEG-2 Format Profile: Audio portion profiling: MPEG-1 L2, MPEG-2 L2 **AV Audio Profile 3** Profiles **MPEG_PS_ NTSC** and **MPEG_PS_ PAL**	M	C	DMS, DMP	7.7.17.1 MPEG-1 and MPEG-2 Audio Layer 2 Stream must match the provisions about MPEG Audio in 1.2 General Specifications of Presentation Data, 5.1 Definition of VOB, 5.3.3 Audio Pack and 5.5.2.3 MPEG audio in [47]. Main characteristics of MPEG Audio Layer 2 stream are: **Stream ID in MPEG-2 System** 0b1100_000* for main audio stream 0b1101_000* for MEPG-2 extension audio stream 0b110x_000* : * corresponds to audio stream number. (x is 0 or 1 depending if this is main or extension stream) *=0 for main language. *=1 corresponds to auxiliary language.(defined in Table 5.3.3-3 : Audio packet of MPEG audio in [47]). **Sampling Rate** 48 kHz. (defined in Table 5.5.2.3-1 : Restricted Items for MPEG coding of audio in [47]) **Protection** CRC check always on. (defined in Table 5.5.2.3-1 : Restricted Items for MPEG coding of audio in [47]) If bad (invalid) CRC is detected, DMP can mute the audio or deploy manufacturer-specific error-concealment techniques. **Audio Channel Mode** ➢ 2 channel (stereo, dual mono) ➢ 1 channel (mono) Changing audio channels among mono, dual-mono and stereo is allowed, (defined in Table 5.5.2.3-1: Restricted Items for MPEG coding of audio and Table C-4 : Relation between Presentation Data of Audio and the Player in [47]). **Audio Bit rate** ➢ ***MPEG-1 Audio Layer 2*** 1 channel (mono) : from 64 kbps to 192 kbps 2 channel (stereo, dual mono) from 64 kbps to 384 kbps	[43] [44] [47]	

Table 7-7 AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
				➢ ***MPEG-2 Audio Layer 2*** *Main stream:* ➢ 1 channel (mono) : from 64 kbps to 192 kbps ➢ 2 channel (stereo, dual mono) from 64 kbps to 384 kbps *Extension stream:* Up to 528 kbps (defined in Table 5.5.2.3-1 : Restricted Items for MPEG coding of audio in [47]) **Emphasis** Always no emphasis. (defined in Table 5.5.2.3-1 : Restricted Items for MPEG coding of audio in [47]) **Private bit** 0 (defined in Table 5.5.2.3-1 : Restricted Items for MPEG coding of audio in [47]) **Number of multilingual channels** 0 (defined in Table 5.5.2.3-1 : Restricted Items for MPEG coding of audio in [47]) **mc_prediction_on** 0(use of prediction excluded). (defined in Table 5.5.2.3-1 : Restricted Items for MPEG coding of audio in [47]) **Number of bits reserved for "dynamic_range_control" in ancillary data** 16. (defined in Table 5.5.2.3-1 : Restricted Items for MPEG coding of audio in [47]) **PTS in Audio Packet** PTS must be described in every audio packet in which the first sample of audio frame is included. PTS[32] must be set to zero. .(defined in Table 5.3.3-3 Audio Packet of MPEG Audio in [47]) **Audio Packet Header** Packet Header must match the provisions defined in Table 5.3.3-3 : Audio packet of MPEG Audio in [47]		

Table 7-7 AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.18 MF Baseline AV MPEG2 Format Profile: IFO File Format	M	A	DMS, DMP	7.7.18.1 If the IFO file is used, then the Format of an IFO file must conform to the DVD-VR IFO format except for the following provision: Any given IFO file must include the information only for one piece of the corresponding to this IFO file DLNA PS-formatted content. The maximum size of the IFO file must not exceed 512 KB	[47]	For media management-related guideline entry, see 7.3.38.
7.7.19 MF Baseline AV MPEG2 Format Profile: Invalid Field in IFO File	M	A	DMP DMS	7.7.19.1 While interpreting the information in the IFO file, rendering endpoints must ignore the “VMG_ID”, “ETKI” and “CPSI” values in the IFO file. 7.7.19.2 If a serving endpoint exposes an IFO file, then it must fill in the “VMG_ID”, “ETKI” and “CPSI” fields in the IFO file with the value, “reserved”.	[47]	These fields are related to contents protection and must not be used over the network.
7.7.20 MF Baseline AV MPEG2 Format Profile: IFO File MIME Type	M	A	DMS	7.7.20.1 The following MIME-Type must be used in indicating the IFO file resource: application/octet-stream.	[47]	This MIME type can be used when a rendering endpoint retrieves an IFO file using HTTP **GET** method.

Optional AV Media Class Formatting Guidelines

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
General Format Guidelines for Transport Stream						
7.7.21 MF Baseline[a]/ Optional AV Media Format Profile: MPEG-2 MIME Type Definition	M	A		7.7.21.1 MIME type “video/mpeg” must be used for this media format.		

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.22 MF Baseline[aa]/ Optional AV Media Format Profile: MPEG-2 Transport Stream Profiling	M	L	DMS	7.7.22.1 The Transport Streams (TS) must be profiled according to the video format-specific requirements listed in the guideline entries below	[41] [42] [43] [51] [54] [44]	
	M	L	DMP	7.7.22.2 Rendering endpoints must support (render) the MPEG-2 Transport Stream (TS) profiled according to all video format-specific requirements listed in this Table. Rendering endpoints (e.g., DMP_B) must support TS profiled by the regional transport stream profile specification according to the region of rendering endpoints' intended deployment..	[41] [42] [43] [51] [54] [44]	As an example, consider DMP_B sold in Europe. MPEG-2 TS must be profiled according to **MPEG_TS_SD_EU**,format profile.
Common Format Specific Requirements						
7.7.23 MF Baseline[a]/ Optional AV Media Format Profile: System Stream Specification Profiles **MPEG_TS_SD_NA**, **MPEG_TS_SD_NA_T**, **MPEG_TS_HD_NA**, **MPEG_TS_HD_NA_T**, **MPEG_TS_SD_EU**, **MPEG_TS_SD_EU_T**, **MPEG_TS_SD_KO**, **MPEG_TS_SD_KO_T**, **MPEG_TS_HD_KO**, and **MPEG_TS_HD_KO_T**	M	L	DMP	7.7.23.1 Main characteristics of MPEG-2 system stream are **System** System Stream is Full or Partial Single Program Transport Stream (SPTS). It must contain only one program from the original broadcasted TS according to the PAT and PMT tables. **Number of Programs** SPTS: Single program Transport Stream **Packet Size** 188 bytes per transport packet **PSI Information** PAT and PMT are required to be present in the stream. It must tolerate any other tables. PSI insertion interval should be decided by the implementers.	[41] [42] [43] [51] [54] [44]	A partial SPTS can be created from MPTS by removing packets corresponding to other programs, or from another SPTS stream (by removing some components). A full or partial SPTS can have multiple audio and video components in it. The 188 byte transport stream packet specified in this guideline is always preceded by a 32 bit timestamp making a 192 byte packet, as described in 7.7.26.1. Full or partial SPTS is fully compliant with [41].

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
	M	L	DMS	7.7.23.2 Main characteristics of MPEG-2 system stream are **System** System Stream is Full or Partial Single Program Transport Stream (SPTS). It must contain only one program from the original broadcasted TS according to the PAT and PMT tables. **Number of Programs** SPTS: Single program Transport Stream **Packet Size** 188 bytes per transport packet **PSI Information** PAT and PMT are required. PSI insertion interval should be decided by the implementers. Serving endpoints may choose to send additional PSI information.	[41] [42] [43] [51] [54] [44]	A full or partial SPTS can have multiple audio and video components in it. The 188 byte transport stream packet specified in this guideline is always preceded by a 32 bit timestamp making a 192 byte packet, as described in 7.7.26.1.
7.7.24 MF Baseline[a]/ Optional AV Media Format Profile: Rendering Requirements Profiles **MPEG_TS_SD_NA**, **MPEG_TS_SD_NA_T**, **MPEG_TS_HD_NA**, **MPEG_TS_HD_NA_T**, **MPEG_TS_SD_EU**, **MPEG_TS_SD_EU_T**, **MPEG_TS_SD_KO**, **MPEG_TS_SD_KO_T**, **MPEG_TS_HD_KO**, and **MPEG_TS_HD_KO_T**	M	A	DMP	7.7.24.1 A rendering endpoint must accept (tolerate) but not necessarily render all audio, video and data components that belong to the (selected) program in a full or partial SPTS according to the PAT/PMT tables. At a minimum, an AV-supporting DMP_B must render one elementary video stream and one corresponding elementary audio stream as present in TS, following additional provisions in this Table.	[41] [42] [43] [51] [54] [44]	AV device supporting AV media class content, must support rendering of both, audio and video components.

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.25 MF Baseline[a]/ Optional AV Media Format Profile: Video Elementary Stream Specification Profiles: **MPEG_TS_SD_NA**, **MPEG_TS_SD_NA_T**, **MPEG_TS_SD_EU**, **MPEG_TS_SD_EU_T**, **MPEG_TS_SD_KO**, and **MPEG_TS_SD_KO_T**	M	L		7.7.25.1 Main characteristics of MPEG-2 video stream are **Profile:** MP@ML **Chroma** 4:2:0 **Video bit rate** ➢CBR: Equal to or less than 15 Mbps ➢VBR: maximum bit rate equal to or less than 15 Mbps	[41] [42] [43] [51] [54]	Some profiles may also utilize MP@HL MPEG-2 profile, as will be described in the guideline entries in this Table.
7.7.26 MF Baseline[a]/ Optional AV Media Format Profile Time Stamped TS Packet Format	M	R	DMP, DMS	7.7.26.1 When TS packet is transmitted by HTTP, each MPEG-2 TS packet shall be preceded by 32bits Time stamp making 192 Bytes packet (4 Byte timestamp + 188 Byte transport packet) as defined in [65]. This 32 bit field is a 27MHz clock binary counter value to control the relative input timing to the decoder of the following transport packet. The 27MHz clock is synchronized to the MPEG system clock but this counter value may not be equal to the STC counter value. (i.e. they may have offset)If serving endpoint is not capable of generating the timestamp, serving endpoint may put 0x00000000 to the timestamp field of the all packets in the stream. (Either all packets in the stream have valid timestamp or all packets have zero-valued timestamp)If rendering endpoint observes zero-valued timestamp in the first 2 packets of the stream, rendering endpoint treats the timestamp as not available for the stream. When timestamp is provided, the accuracy of the timestamp is recommended to be less or equal to +-500nS. Corresponding Profile IDs have "_T" appended as part of their profile name (e.g. MPEG_TS_SD_NA_T etc.) to indicate the content stream has valid timestamp values.	[65]	DMP may ignore the timestamp. TTS timestamp and PCR are different in the following aspects: • PCR consists of 9bit 27MHz MPEG STC extension part and 33bit 90 kHz base part. TTS timestamp is 32bit binary 27MHz counter. • PCR may have discontinuity. TTS timestamp is continuous for the duration of the stream. • PCR is present with an interval of 100ms or less. TTS timestamp present in every TS packet.

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.27 MF Baseline[a]/ Optional AV Media Format Profile. HTTP alignment with TS packet structures Profiles: **MPEG_TS_SD_NA**, **MPEG_TS_SD_NA_T**, **MPEG_TS_HD_NA**, **MPEG_TS_HD_NA_T**, **MPEG_TS_SD_EU**, **MPEG_TS_SD_EU_T**, **MPEG_TS_SD_KO**, **MPEG_TS_SD_KO_T**, **MPEG_TS_HD_KO**, and **MPEG_TS_HD_KO_T**	M		DMS	7.7.27.1 The first byte of an HTTP response to a client carrying the media content in its entity body must be aligned to the TS packet boundary, unless **RANGE** header field is used in this HTTP request, and the specified starting index is not aligned to the TS packet boundary.		
US Region Specific TS Profiling Requirements: MPEG_TS_SD_NA, MPEG_TS_SD_NA_T, MPEG_TS_HD_NA, and MPEG_TS_HD_NA_T Profiles						
7.7.28 MF Baseline[a]/ Optional AV Media Format Profile: US region TS Profile Definition Profiles **MPEG_TS_SD_NA**, **MPEG_TS_SD_NA_T**, **MPEG_TS_HD_NA**, and **MPEG_TS_HD_NA_T**	M	A	DMS, DMP	7.7.28.1 The following must be used for this DLNA media format profile **For SD content** Profile = MPEG_TS_SD_NA **For HD content** Profile = MPEG_TS_HD_NA DLNA MF profileID “MPEG_TS_SD_NA” must be used to indicate the content of this media format profile for standard definition (SD) content. DLNA MF profileID “MPEG_TS_HD_NA” must be used to indicate the content of this media format profile for high definition (HD) content.		

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.29 MF Baseline[a]/ Optional AV Media Format Profile: MPEG-2 AV Stream Profile **MPEG_TS_SD_NA**, **MPEG_TS_SD_NA_T**	M	L	DMP,DMS	7.7.29.1 **System bit rate** SPTS: Up to 19.3927 Mbps	[51]	
7.7.30 MF Baseline[a]/ Optional AV Media Format Profile: MPEG-2 Closed Caption Stream Profiles **MPEG_TS_SD_NA**, **MPEG_TS_SD_NA_T**, **MPEG_TS_HD_NA**, and **MPEG_TS_HD_NA_T**	M	R	DMP	7.7.30.1 A rendering endpoint must tolerate the CEA-708-B closed captioning. **Bit rate:** 9.6 kbps	[51]	

Optional AV Media Class Formatting Guidelines (Continued)

<table>
<tr><th>Name</th><th>M/S/</th><th>A/C/F</th><th>Dev Class</th><th>Description</th><th>Ref #</th><th>Comments</th></tr>
<tr><td>7.7.31
MF Baseline[a]/ Optional AV Media Format Profile: MPEG-2 Video Format Profile MPEG_TS_SD_NA, MPEG_TS_SD_NA_T</td><td>M</td><td>C</td><td>DMS, DMP</td><td>7.7.31.1
Video Encoding Parameters:
.
MPEG_TS_SD_NA, MPEG_TS_SD_NA_T
RESOLUTION / ASPECT RATIO / FIELD (INTERLACED) OR FRAME (PROGRESSIVE) RATE
➢720*480 16:9 29.97P
➢720*480 16:9 59.94I
➢720*480 4:3 29.97P
➢720*480 4:3 59.94I
➢704*480 16:9 59.94I & 60I
➢704*480 16:9 29.97P & 30P
➢704*480 16:9 23.976P & 24P
➢704*480 4:3 59.94I & 60I
➢704*480 4:3 29.97P & 30P
➢704*480 4:3 23.976P & 24P
➢640*480 4:3 59.94I & 60I
➢640*480 4:3 29.97P & 30P
➢640*480 4:3 23.976P & 24P
➢640*480 SQUARE 59.94I & 60I
➢640*480 SQUARE 29.97P & 30P
➢640*480 SQUARE 23.976P & 24P
➢544*480 16:9 59.94I
➢544*480 4:3 29.97P
➢544*480 4:3 59.94I
➢480*480 16:9 59.94I
➢480*480 4:3 29.97P
➢480*480 4:3 59.94I
➢352*480 16:9 59.94I
➢352*480 4:3 29.97P
➢352*480 4:3 59.94I</td><td>[41]
[42]
[43]
[51]
[61]
[62]</td><td>60i means interlaced sequence of 30 fps frame rate</td></tr>
</table>

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.32 MF Baseline[a]/ Optional AV Media Format Profile: ATSC EDTV Stream Format Profile **MPEG_TS_SD_NA**, **MPEG_TS_SD_NA_T**	M	C	DMS, DMP	7.7.32.1 Main characteristics of MPEG-2 video stream are: **Profile:** MP@HL **Chroma:** 4:2:0 **Video bit rate** ➢CBR: less than or equal to (19.3927Mb/s minus the sum of audio, close caption data, other SI tables and TS overhead) ➢VBR: maximum bit rate less than or equal to (19.3927Mb/s minus the sum of audio, close caption data other SI tables and TS overhead)	[41] [42] [43] [51]	This is an additional profiling of MPEG_TS_SD_NA format profile that specifies different MPEG-2 Profile (i.e. MP@HL).
7.7.33 MF Baseline[a]/ Optional AV Media Format Profile: ATSC EDTV Video Format Profile **MPEG_TS_SD_NA**, **MPEG_TS_SD_NA_T**	M	C	DMP,DMS	7.7.33.1 **Video Encoding Parameters:** **Additional Parameters for MPEG_TS_SD_NA, MPEG_TS_SD_NA_T** RESOLUTION / ASPECT RATIO / FIELD (INTERLACED) OR FRAME (PROGRESSIVE) RATE ➢704*480 — 16:9 — 59.94P & 60P ➢704*480 — 4:3 — 59.94P & 60P ➢640*480 — 4:3 — 59.94P & 60P ➢640*480 — SQUARE — 59.94P & 60P	[41] [42] [43] [51]	

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.34 MF Baseline[a]/ Optional AV Media Format Profile: MPEG-2 Audio Portion Profiling: **AC-3** Profile **MPEG_TS_SD_NA**, **MPEG_TS_SD_NA_T**, **MPEG_TS_HD_NA**, **MPEG_TS_HD_NA_T**	M	R/L	DMP	7.7.34.1 Main characteristics of ATSC AC-3 audio stream are: **Sampling Rate** 48 KHz **Content audio channel modes:** Rendering endpoints must tolerate the following AC-3 formats ➢ Mono (1/0) ➢ Stereo (2/0) ➢ Multi-channels (3/0) ➢ Multi-channels (2/1) ➢ Multi-channels (3/1) ➢ Multi-channels (2/2) ➢ Multi-channels (3/2) **Output Audio Channel modes:** ➢ 2 channels ➢ 1 channel **Audio bit rate** • Main audio service up to 448 kbps per one stream. The combined bit rate of a main service and an associated service which are intended to be decoded simultaneously must be up to 576 kbps	[45] [51]	32 KHz and 44.1 KHz AC-3 audio sampling rates were removed from the ATSC A/53B Annex B normative standard. 576 kbps is the maximum ATSC AC-3 audio bit rate instead of 640 kbps per ATSC A/53B Annex B normative standard
7.7.35 MF Optional AV Media Format Profile: **MPEG_TS_HD_NA**, **MPEG_TS_HD_NA_T**	O		DMS DMP	7.7.35.1 The contents of an AV media class may be profiled according to the optional DLNA media format profiles, **MPEG_TS_HD_NA** and **MPEG_TS_HD_NA_T**, as indicated in the entries 7.7.36 through 7.7.38.		
7.7.36 MF Optional AV Media Format Profile: MPEG-2 AV Stream Profile **MPEG_TS_HD_NA**, **MPEG_TS_HD_NA_T**	M	L	DMP DMS	7.7.36.1 **System bit rate (minimum requirements)** SPTS: CBR or VBR up to 19.3927 Mbps	[51]	Though **MPEG_TS_HD_NA** and **MPEG_TS_HD_NA_T** are optional media format profiles, indicated profiling is mandatory in case this media profile is used.

Optional AV Media Class Formatting Guidelines (Continued)

<table>
<tr><th>Name</th><th>M/S/</th><th>A/C/F</th><th>Dev Class</th><th>Description</th><th>Ref #</th><th>Comments</th></tr>
<tr><td>7.7.37
MF Optional AV Media Format Profile: MPEG-2 Stream Format Profile
MPEG_TS_HD_NA, MPEG_TS_HD_NA_T</td><td>M</td><td>L</td><td>DMP
DMS</td><td>7.7.37.1 Main characteristics of MPEG-2 video stream are:
Profile: MP@HL
Chroma 4:2:0
Video bit rate
➢CBR: less than or equal to (19.3927Mb/s minus the sum of audio, close caption data, other SI tables and TS overhead)
➢VBR: maximum bit rate less than or equal to (19.3927Mb/s minus the sum of audio, close caption data other SI tables and TS overhead)</td><td>[41]
[42]
[43]
[51]</td><td>Though MPEG_TS_HD_NA and MPEG_TS_HD_NA_T are optional media format profiles, indicated profiling is mandatory in case this media profile is used.</td></tr>
<tr><td>7.7.38
MF Optional AV Media Format Profile: MPEG-2 Video Format Profile
MPEG_TS_HD_NA, MPEG_TS_HD_NA_T</td><td>M</td><td>L</td><td>DMP
DMS</td><td>7.7.38.1
Video Encoding Parameters:
MPEG_TS_HD_NA, MPEG_TS_HD_NA_T
RESOLUTION / ASPECT RATIO / FIELD (INTERLACED) OR FRAME (PROGRESSIVE) RATE
➢1920*1080 16:9 59.94I & 60I
➢1920*1080 16:9 29.97P & 30P
➢1920*1080 16:9 23.976P & 24P
➢1920*1080 SQUARE 59.94I & 60I
➢1920*1080 SQUARE 29.97P & 30P
➢1920*1080 SQUARE 23.976P & 24P
➢1280*720 16:9 59.94P & 60P
➢1280*720 16:9 29.97P & 30P
➢1280*720 16:9 23.976P & 24P
➢1280*720 SQUARE 59.94P & 60P
➢1280*720 SQUARE 29.97P & 30P
➢1280*720 SQUARE 23.976P & 24P
➢1440*1080 16:9 29.97P & 30P
➢1440*1080 16:9 23.976P & 24P
➢1440*1080 16:9 59.94I & 60I
➢1280*1080 16:9 29.97P & 30P
➢1280*1080 16:9 23.976P & 24P
➢1280*1080 16:9 59.94I & 60I</td><td>[41]
[42]
[43]
[51]
[61]
[62]</td><td>Though MPEG_TS_HD_NA and MPEG_TS_HD_NA_T are optional media format profiles, indicated profiling is mandatory in case this media profile is used.</td></tr>
</table>

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
Korean Region Specific TS Profiling Requirements. MPEG_TS_SD_KO, MPEG_TS_SD_KO_T, MPEG_TS_HD_KO, and MPEG_TS_HD_KO_T profiles.						
7.7.39 MF Baseline[a]/ Optional AV Media Format Profile: Korea Region TS Profile Definition Profiles **MPEG_TS_SD_KO**, **MPEG_TS_SD_KO_T**, **MPEG_TS_HD_KO**, and **MPEG_TS_HD_KO_T**	M	A	DMP, DMS	7.7.39.1 The following must be used for this DLNA media format profile **For SD content** Profile = MPEG_TS_SD_KO **For HD content** Profile = MPEG_TS_HD_KO DLNA MF profileID "MPEG_TS_SD_KO" must be used to indicate the content of this media format profile for SD content. DLNA MF profileID "MPEG_TS_HD_KO" must be used to indicate the content of this media format profile for HD content.		
7.7.40 MF Baseline[a]/ Optional AV Media Format Profile: MPEG-2 AV Stream Profile **MPEG_TS_SD_KO** and **MPEG_TS_SD_KO_T**	M	L	DMP,DMS	7.7.40.1 **System bit rate** SPTS: Up to 19.3927 Mb/s	[51]	
7.7.41 MF Baseline[a]/ Optional AV Media Format Profile: Closed caption Profile **MPEG_TS_SD_KO**, **MPEG_TS_SD_KO_T**, **MPEG_TS_HD_KO**, and **MPEG_TS_HD_KO_T**	M	R	DMP, DMS	7.7.41.1 A Rendering endpoint must tolerate the CEA-708-B closed captioning **Bit rate:** 9.6 kbps	[51]	

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.42 MF Baseline[a]/ Optional AV Media Format Profile: MPEG-2 Audio Portion Profiling **AC-3** Profile **MPEG_TS_SD_KO**, **MPEG_TS_SD_KO_T**, **MPEG_TS_HD_KO**, and **MPEG_TS_HD_KO_T**	M	L	DMP, DMS	7.7.42.1 Main characteristics of ATSC AC-3 audio stream are **Sampling Rate** 48 KHz **Content audio channel modes:** DMP must tolerate the following AC-3 formats ➢ Mono (1/0) ➢ Stereo (2/0) ➢ Multi-channels (3/0) ➢ Multi-channels (2/1) ➢ Multi-channels (3/1) ➢ Multi-channels (2/2) ➢ Multi-channels (3/2) **Output Audio Channel modes:** ➢ 2 channels ➢ 1 channel **Audio bit rate** Main audio service up to 448 kbps per one stream. The combined bit rate of a main service and an associated service which are intended to be decoded simultaneously must be up to 576 kbps	[45] [51]	32 KHz and 44.1 KHz AC-3 audio sampling rates were removed from the ATSC A/53B Annex B normative standard. 576 kbps is the maximum ATSC AC-3 audio bit rate instead of 640 kbps per ATSC A/53B Annex B normative standard

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.43 MF Baseline[a]/ Optional AV Media Format Profile: Video format Profile **MPEG_TS_SD_KO** and **MPEG_TS_SD_KO_T**	M	L	DMP, DMS	7.7.43.1 **Video Encoding Parameters** **MPEG_TS_SD_KO, MPEG_TS_SD_KO_T** RESOLUTION / ASPECT RATIO / FIELD (INTERLACED) OR FRAME (PROGRESSIVE) RATE ➢704*480 — 16:9 — 29.97P & 30P ➢704*480 — 16:9 — 23.976P & 24P ➢704*480 — 16:9 — 59.94I & 60I ➢704*480 — 4:3 — 29.97P & 30P ➢704*480 — 4:3 — 23.976P & 24P ➢704*480 — 4:3 — 59.94I& 60I ➢640*480 — 4:3 — 29.97P & 30P ➢640*480 — 4:3 — 23.976P & 24P ➢640*480 — 4:3 — 59.94I & 60I	[41] [42] [43] [51]	
7.7.44 MF Optional AV Media Format Profile: **MPEG_TS_HD_KO** and **MPEG_TS_HD_KO_T**	O		DMS, DMP	7.7.44.1 The contents of an AV media class may be profiled according to the optional DLNA media format profiles, **MPEG_TS_HD_KO** and **MPEG_TS_HD_KO_T**, as indicated in the entries 7.7.45 through 7.7.47.		

Optional AV Media Class Formatting Guidelines (Continued)

<table>
<tr><th>Name</th><th>M/S/</th><th>A/C/F</th><th>Dev Class</th><th>Description</th><th>Ref #</th><th>Comments</th></tr>
<tr><td>7.7.45
MF Optional AV Media Format Profile: Video stream specification Profile MPEG_TS_HD_KO and MPEG_TS_HD_KO_T</td><td>M</td><td>L</td><td>DMP, DMS</td><td>7.7.45.1
System bit rate (minimum requirements)
SPTS: CBR/VBR up to 19.3927 Mbps</td><td>[51]</td><td>Though MPEG_TS_HD_KO and MPEG_TS_HD_KO_T are optional media format profiles, indicated profiling is mandatory in case this media profile is used.</td></tr>
<tr><td>7.7.46
MF Optional AV Media Format Profile: Video Stream Format Profile MPEG_TS_HD_KO and MPEG_TS_HD_KO_T</td><td>M</td><td>L</td><td>DMP, DMS</td><td>7.7.46.1 Main characteristics of MPEG-2 video stream are
Profile: MP@HL
Chroma 4:2:0
Video bit rate
➢ CBR: less than or equal to (19.3927Mb/s minus the sum of audio, close caption data, other SI tables and TS overhead)
➢ VBR: maximum bit rate less than or equal to (19.3927Mb/s minus the sum of audio, close caption data other SI tables and TS overhead)</td><td>[41]
[42]
[43]
[51]</td><td>Though MPEG_TS_HD_KO and MPEG_TS_HD_KO_T are optional media format profiles, indicated profiling is mandatory in case this media profile is used.</td></tr>
<tr><td>7.7.47
MF Optional AV Media Format Profile: Video format Profile MPEG_TS_HD_KO and MPEG_TS_HD_KO_T</td><td>M</td><td>L</td><td>DMP</td><td>7.7.47.1
Video Encoding Parameters
MPEG_TS_HD_KO, MPEG_TS_HD_KO_T
<table><tr><th>RESOLUTION</th><th>ASPECT RATIO</th><th>FIELD (INTERLACED) OR FRAME (PROGRESSIVE) RATE</th></tr><tr><td>➢1920*1080</td><td>16:9</td><td>59.94I & 60I</td></tr><tr><td>➢1280*720</td><td>16:9</td><td>59.94P & 60P</td></tr><tr><td>➢704*480</td><td>16:9</td><td>59.94P & 60P</td></tr><tr><td>➢704*480</td><td>4:3</td><td>59.94P & 60P</td></tr><tr><td>➢640*480</td><td>4:3</td><td>59.94P & 60P</td></tr></table></td><td>[41]
[42]
[43]
[51]</td><td>Though MPEG_TS_HD_KO and MPEG_TS_HD_KO_T are optional media format profiles, indicated profiling is mandatory in case this media profile is used.</td></tr>
</table>

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
DVB-European Region Specific TS Profiling Requirements. MPEG_TS_SD_EU and MPEG_TS_SD_EU_T Profiles						
7.7.48 MF Baseline[a]/ Optional AV Media Format Profile: Europe Region: TS Profile Definition Profile **MPEG_TS_SD_EU** and **MPEG_TS_SD_EU_T**	M	A	DMP, DMS	7.7.48.1 The following dlna-profile must be used for this media format **For SD content** Profile = MPEG_TS_SD_EU DLNA MF profileID "MPEG_TS_SD_EU" must be used to indicate the content of this media format profile for SD content.		
7.7.49 MF Baseline[a]/ Optional AV Media Format Profile: Europe Region: Compliance Profile **MPEG_TS_SD_EU** and **MPEG_TS_SD_EU_T**	M	L	DMP, DMS	7.7.49.1 Serving and rendering endpoints must comply with the following: • Section 4.1 in **[54]** defining the use of **[41]** for IRDs and bit streams in DVB. • Section 5.1 in **[54]** defining the use of **[42]** for 25 Hz SDTV IRDs and bit streams in DVB. • Section 6 in **[54]** defining the use of **[43]** and **[44]** for IRDs and bit streams in DVB. • All requirements for the MPEG_TS_SD_EU profile defined in this document that further specify, limit or otherwise modify the requirements for 25 Hz SDTV IRDs and bit streams in DVB as defined in **[54]**.	**[41]** **[42]** **[43]** **[44]** **[54]**	30 Hz systems and HDTV are out of scope of this guideline
7.7.50 MF Baseline[a]/ Optional AV Media Format Profile: Subtitles for TS Profile **MPEG_TS_SD_EU** and **MPEG_TS_SD_EU_T**	O	R	DMS	7.7.50.1 Serving endpoints may output DVB subtitles as specified in **[56]**.	**[56]**	
	M	R	DMP	7.7.50.2 Rendering endpoints must tolerate but not necessarily render DVB subtitles as specified in **[56]**.	**[56]**	
	S	A	DMP	7.7.50.3 A rendering endpoint should be capable of rendering DVB subtitles as specified in **[56]** if they are present.	**[56]**	If DVB subtitles are not used in the country in which the device is sold, this recommendation is not applicable.

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.51 MF Baseline[a]/ Optional AV Media Format Profile: Teletext for TS Profile **MPEG_TS_SD_EU** and **MPEG_TS_SD_EU_T**	O	R	DMS	7.7.51.1 Serving endpoints may output DVB teletext as specified in **[58]**.	**[58]**	
	M	R	DMP	7.7.51.2 Rendering endpoints must tolerate but not necessarily render teletext as specified in **[58]**.	**[58]**	
7.7.52 MF Baseline[a]/ Optional AV Media Format Profile: DVB SDTV Systems Profile **MPEG_TS_SD_EU** and **MPEG_TS_SD_EU_T**	M	R	DMS, DMP	7.7.52.1 Main characteristics of the MPEG-2 European DVB Full Transport Stream are: TS format with PSI and SI tables as specified in section 4.1 of **[54]**, **[55]**. Other program-related data may be present as per **[60]**.	**[54]** **[55]** **[60]**	The intent of this guideline is to ensure that TS complies with relevant DVB specifications. Further guidelines are listed in this table.
	M	F	DMS, DMP	7.7.52.2 A partial SPTS format is a TS with gaps of variable length between the TS packets. It must contain only one program from the original broadcast TS. A partial SPTS must carry at least the following PSI and SI tables as defined in **[41]**, **[54]**, and **[55]**: • PAT and PMT. • SIT and the corresponding partial TS descriptor. • DIT whenever there is a discontinuity in the SI information.	**[41]** **[54]** **[55]** **[60]**	Partial SPTS is fully MPEG-2 compliant according to **[41]**, amended by 7.7.52.4.
	O	A	DMP	7.7.52.3 A rendering endpoint may process a DIT as defined in **[55]** in a partial SPTS.	**[55]**	
	S	A	DMS	7.7.52.4 A serving endpoint is strongly recommended to pass on or insert into a partial SPTS all the SI tables as defined in **[55]** that were part of the original broadcast TS as well as all audio, video and data components that belong to the (selected) program according to the PAT/PMT tables, and CAT, if need be. This includes components that contain DVB defined data that belong to that program such as teletext as defined in **[58]**, other VBI data as defined in **[57]**, object carrousels as defined in **[59]** **[73]**, or MHP data as defined in **[60]**.	**[55]** **[57]** **[58]** **[59]** **[73]** **[60]**	Even though DRM is out of scope of this version, it should be allowed to insert the CAT table to accommodate a DMP that is capable of decryption

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
	M	A	DMP	7.7.52.5 A rendering endpoint must tolerate all audio, video and data components that belong to the (selected) program in a full or partial SPTS according to the PAT/PMT tables. This includes components that contain DVB defined data that belong to that program such as teletext as defined in [58], other VBI data as defined in [57], object carrousels as defined in [59] [73], or MHP data as defined in [60] and others.	[55] [57] [58] [59] [73] [60]	
	M	A	DMP	7.7.52.6 When the SI tables of a partial SPTS refer to programs that are not present in the partial SPTS or otherwise conflict with data in the SIT table of that partial SPTS, the DMP must ignore this SI data.		
	M	A	DMP	7.7.52.7 A rendering endpoint that is capable of rendering both audio and video must be able to render a full or partial SPTS containing a program with one Video ES up to 15 Mb/s and one stereo (2/0) Audio ES.		

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.53 MF Baseline[a]/ Optional AV Media Format Profile: **MPEG_TS_SD_EU** and **MPEG_TS_SD_EU_T** Profile Mandatory Audio Portion Profiling: **MPEG-1/2 audio**	M	R	DMS, DMP	7.7.53.1 Main characteristics of the MPEG-2 TS Format / European DVB Profile for MPEG-1 or MPEG-2 audio streams (i.e. mandatory audio portion profiling), according to Section 6 of **[54]** defining the use of **[43]** and **[44]**: **Coding** ➢ MPEG-1 Layer 1 ➢ MPEG-1 Layer 2 A serving endpoint may output, and a rendering endpoint must be capable of rendering a joint stereo encoded audio stream. No mc-prediction. **MPEG-1 backwards compatible multi-channel mode:** ➢ MPEG-2 Layer 2 mc (multi-channel), dematrix procedure 0, 1 or 2 A serving endpoint must output one of and a DMP must be capable of rendering all of the following Content Audio Channel Modes: ➢ 1/0: single channel ➢ 1/0+1/0: dual channel ➢ 2/0: stereo A serving endpoint may output and a rendering endpoint must accept but not necessarily render the following multi-channel Content Audio Channel Modes, that optionally include a Low Frequency Enhancement (LFE) channel: ➢ 2/2 ➢ 2/1 ➢ 3/2 ➢ 3/1 ➢ 3/0 ➢ 3/0+2/0 ➢ 2/0+2/0 ➢ (1/0+1/0)+2/0 ➢ 1/0+2/0 A serving endpoint may output and a rendering endpoint must accept but not necessarily render audio data contained in an extension stream and ancillary data fields of the audio stream No multilingual channels in multi-channel mode CRC check must be included	**[43]** **[44]** **[54]**	Serving endpoints must output one of the three basic content audio channel modes. The optional multi-channel modes use the ancillary data fields and optionally an extension stream, thereby preserving backwards compatibility to basic stereo decoders. Decoding of all multi-channel modes is optional.

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
				Sample rates: ➢ 32 kHz ➢ 44.1 kHz ➢ 48 kHz **Bitrates:** ➢ Layer 1: from 32 to 448 kbps ➢ Layer 2: from 32 to 384 kbps **Extension stream (optional):** From 0 to 682 kbps **No emphasis**		
7.7.54 MF Baseline[a]/ Optional AV Media Format Profile: **MPEG_TS_SD_EU** and **MPEG_TS_SD_EU_T** Profile Mandatory Audio Portion Profiling: **MPEG-1/2 audio**	M	A	DMS, DMP	7.7.54.1 A serving endpoint may output and a rendering endpoint must be capable of rendering the following additional sample rates for secondary sound services: ➢ 16 kHz ➢ 22.05 kHz ➢ 24 kHz	[54]	European broadcasters expect DMPs to support this.

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.55 MF Baseline[a]/ Optional AV Media Format Profile: **MPEG_TS_SD_EU** and **MPEG_TS_SD_EU_T** Profile Optional Audio Portion Profiling: **AC-3**	O		DMS, DMP	7.7.55.1 The audio portion in the contents of an AV media class may be profiled according to the Optional Audio Portion Profiling, **AC-3**, as indicated in the entries 7.7.55.2 and 7.7.56.1.		
	M	R	DMS, DMP	7.7.55.2 Main characteristics of the MPEG-2 TS Format / European DVB Profile AC-3 audio stream (i.e., optional audio portion profiling) are: The coding and decoding of an AC-3 elementary stream is based upon **[53]**: ITU-R Recommendation BS.1196-E - Annex 2. However, Appendix 1 to Annex 2 of **[53]** should be disregarded, as it is not applicable to the DVB system **Sample rates:** ➢ 32 kHz ➢ 44.1 kHz ➢ 48 kHz A serving endpoint may output and a rendering endpoint must accept but not necessarily render the following AC-3 Content Audio Channel Modes that optionally include a Low Frequency Effects (LFE) channel ➢ 1+1 ➢ 1/0 ➢ 2/0 ➢ 3/0 ➢ 2/1 ➢ 3/1 ➢ 2/2 ➢ 3/2 A rendering endpoint must be capable of converting all AC-3 Content Audio Channel Modes, including a Low Frequency Effects (LFE) channel when present, into an appropriate single or dual channel Output Audio Channel Mode that is determined by the rendering endpoint.	**[41]** **[53]** **[54]** **[55]**	If audio in MPEG_TS_SD_EU and MPEG_TS_SD_EU_T media format profiles are profiled according to the AC-3 optional audio portion profiling, then its parameters must follow profile parameter sets as indicated in this guideline entry.

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
				Bit rates: from 32 kbps to 640 kbps AC-3 transmission in a full or partial SPTS according to Annex C of [54]: • The AC-3 packetized elementary stream must conform to the requirements of a user private stream type 1, as described in ISO/IEC 13818-1 [41]. • AC-3 descriptor to distinguish the AC-3 audio stream is defined in [54] [55]. • The AC-3 elementary stream must be byte-aligned within the MPEG-2 full or partial SPTS. This means the initial 8 bits of an AC-3 frame must reside in a single byte, which is carried in the MPEG-2 full or partial SPTS.		
7.7.56 MF Baseline[a]/ Optional AV Media Format Profile: **MPEG_TS_SD_EU** and **MPEG_TS_SD_EU_T** Portion Profiling: **AC-3 Annex C**	M	L	DMP,DMS	7.7.56.1 Serving and rendering endpoints must comply with Annex C of [54].	[54]	[54] only recommends Annex C to support AC-3

Optional AV Media Class Formatting Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.7.57 MF Baseline[a]/ Optional AV Media Format Profile: **MPEG_TS_SD_EU** and **MPEG_TS_SD_EU_T** Video Profile Specification	M	R	XXX	7.7.57.1 Main characteristics of the MPEG-2 TS Format / European DVB SDTV Profile video stream are: A serving endpoint may output and a rendering endpoint must be capable of rendering the following full-screen resolutions: ➢ 720*576 50I ➢ 544*576 50I ➢ 480*576 50I ➢ 352*576 50I ➢ 352*288 50I A serving endpoint may output other resolutions (lower than 720*576) for less than full-screen display. Rendering of these resolutions in full-screen format is optional for a DMP. If a DMP is not capable of rendering these resolutions in full-screen format, it must be able to render them on part of the screen. A serving endpoint may output and a rendering endpoint must be capable of rendering the following aspect ratios: ➢ 4:3 ➢ 16:9 A serving endpoint may output and a rendering endpoint must accept but not necessarily render the following aspect ratios: ➢ 2,21:1 Pan vectors as specified in **[54]** allow a 4:3 DMP to give a full-screen picture of 16:9 content It is recommended that the total bitrate of the set of components, associated PMT and PCR packets for a full or partial SPTS anticipated to be recorded by a consumer, should not exceed 9 Mbpsi.	**[42]** **[54]**	50I means 50 fields per second interlaced rate

a. AV media class profiling by Transport Stream-based profiles is optional for DMPA and mandatory for DMPB types of devices. This requirement is consistent with 7-5 of this guideline specification.

7.8 MEDIA TRANSPORT

This section of the DLNA Home Networked Device Interoperability Guidelines covers the guidelines for media transport.

HTTP CLIENT/SERVER FOR MEDIA STREAMING/RETRIEVAL

The following terminology is used in 7-8 to represent different types of device classes by the role they perform:

- **Rendering Endpoint**: Applies to the device classes that receive content for playback/streaming scenarios. For example, the DMP device class.
- **Serving Endpoint**: Applies to the device classes that serve content for playback/streaming scenarios. For example, the DMS device class that serve content to DMP devices.
- **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.

It should be noted that the guidelines specified in this section apply to DLNA content playback/streaming transactions between DLNA device classes. These guidelines do not specify behavior for non-DLNA device entities. A device class 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.

HTTP Media Transport Guidelines

Table 7-8 HTTP Media Transport Guidelines

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.1 MT Baseline Transport: HTTP	M	L	DMP, DMS	7.8.1.1 A DLNA device must implement HTTP as the mandatory media transport with constraints and extensions defined in subsequent entries of this table. Specifically serving endpoints must implement HTTP servers and rendering endpoints must implement HTTP clients. Guidelines 7.8.12 and 7.8.28 define the HTTP version expectations for HTTP servers and clients.	[24]	DLNA specifies HTTP as the baseline media transport for streaming content.
7.8.2 MT HTTP Play Operation	M	L	DMP	7.8.2.1 Rendering endpoints must use the HTTP **GET** method when using the HTTP transport protocol for media playback/streaming.	[24]	This guideline specifies the normative way to request content.
7.8.3 MT HTTP Stop Operation	S	L	DMP	7.8.3.1 If a rendering endpoint wants to stop the current media stream, it should do so by disconnecting the TCP connection of the HTTP transaction.	[24]	This guideline recommends the way to stop a media stream. Although HTTP clients can technically stall the TCP connection, that technique is not recommended. The recommended technique makes better use of a serving endpoint's platform resources, which may be limited.
7.8.4 MT HTTP Pause Operation	S	L	DMP	7.8.4.1 If a rendering endpoint wants to pause the current media stream, it should do so by one of the following methods: • Disconnecting the TCP connection of the HTTP transaction. • Suspending the reading of data from the TCP connection. Notes: In the latter case, the HTTP server may suddenly disconnect the TCP connection due to a time out. If the connection is disconnected, the HTTP client may use a Seek operation (see guideline 7.8.5) to resume the playback.	[24]	This guideline recommends two ways to pause a media stream. The text under the *Notes* portion of this guideline indicate that a rendering endpoint needs to be prepared for a disconnect scenario and what it can do to recover.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.5 MT HTTP Seek Operation	S	A	DMP	7.8.5.1 If a rendering endpoint wants to perform a seek operation on a stream, it should use the **RANGE** or **TimeSeekRange.dlna.org** header fields.	[24]	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 7.3.11.4 for information on discovering server seek capabilities. See 7.8.28, "MT Baseline Transport: HTTP Client" and 7.8.24, "MT HTTP Time-Based Seek (Server)" guidelines for more information on **RANGE** and **TimeSeekRange.dlna.org** header fields.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.6 MT HTTP Trick Mode Operations	S	A	DMP	7.8.6.1 If a rendering endpoint wants to perform a *Fast Forward Scan* operation (positive play speed greater than 1x), then the rendering endpoint 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.	[24]	*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 rendering 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 rendering endpoint is responsible for choosing the appropriate time-ranges that achieve the desired effect. The third technique works by having the rendering endpoint's HTTP client instruct the serving endpoint's HTTP server to return byte data that is already time-scaled for a particular play speed. See 7.8.22, "MT HTTP Header: RANGE (Server)" , 7.8.24, "MT HTTP Time-Based Seek (Server)", and 7.8.26, "MT HTTP Server-side Trick Mode" guidelines for more information on **RANGE**, **TimeSeekRange.dlna.org,** and **PlaySpeed.dlna.org** header fields.
	S	A	DMP	7.8.6.2 If a rendering endpoint wants to perform a *Slow Forward Scan* operation (positive play speed less than 1x), then the rendering endpoint 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.	[24]	
	S	A	DMP	7.8.6.3 If a rendering endpoint wants to perform a *Fast Backward Scan* operation (negative play speed less than -1x), then the rendering endpoint 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.	[24]	
	S	A	DMP	7.8.6.4 If a rendering endpoint wants to perform a *Slow Backward Scan* operation (negative play speed greater than or equal to -1x), then the rendering endpoint 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.	[24]	

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
	S	A	DMP	7.8.6.5 a rendering endpoint wants to stop a normal playback stream in order to start a *scan operation* playback using the **RANGE** header under conditions where 7.8.9.6 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 rendering endpoint should open a new HTTP connection for the *scan operation*. Please observe that the new HTTP connection may be a persistent connection, which allows the client to issue multiple **GET** requests on a single HTTP connection.	[24]	Note that transitions from *scan operations* to normal playback may be achieved by making open ended (i.e. last-byte-pos value in **RANGE** header is absent) **GET** requests with the **RANGE** option.
	S	A	DMP	7.8.6.6 If a rendering 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 7.8.9.6 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 rendering endpoint should open a new HTTP connection for the *scan operation*. Please observe that the new HTTP connection may be a persistent connection, which allows the client to issue multiple **GET** requests on a single HTTP connection.	[24]	Note that transitions from *scan operations* to normal playback may be achieved by closing the HTTP connection that provides *scan operations* and opening a new HTTP connection for normal playback speed.
7.8.7 MT Graceful Recovery	S	A	DMP	7.8.7.1 Rendering endpoints should not require a hardware reset or a power cycle to return to normal operating conditions after encountering improperly terminated HTTP connections.	N/A	This guideline specifies that a rendering endpoint needs to be able to handle scenarios where an HTTP connection is not properly terminated. Network conditions and/or serving endpoint behavior may cause this scenario to occur. Although a full definition for *graceful recovery* is not provided, a baseline expectation is provided. Essentially, users should not need to reset or power-cycle the device simply because a content transfer was interrupted.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.8 MT DLNA URI Usage	O	C	DMP	7.8.8.1 Rendering endpoints that issue HTTP **GET** 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 rendering endpoint.	[23] [24] [33]	This guideline permits a rendering 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 the 7.3.26, "MM URI Rules", which states that DMS devices must advertise URI values that are URI-escaped. Although a rendering endpoint can implement additional logic for validating a URI, such logic is useful only for interoperation with non-DLNA devices.
7.8.9 MT Valid HTTP Response	M	R	DMS	7.8.9.1 If a serving endpoint receives an HTTP **GET** request, then the serving endpoint must send a valid HTTP response provided the serving endpoint has sufficient *platform resources* (network sockets, stored file in readable state, available tuner hardware, etc.) for sending the response.	[24] [23]	This guideline essentially obligates a serving endpoint to send an HTTP response. It should also be noted that 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, redirections, requests for authorization, HTTP error responses, etc.
	S	R	DMS	7.8.9.2 If a serving endpoint cannot respond to an HTTP **GET** request with content byte data due to the streaming 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).	[24]	This guideline covers the case where the serving endpoint has the resources to send an error response but lacks the resources to send content data. Sending an HTTP error is better than denying the TCP connection request because it explicitly tells the requesting endpoint that content is not available at this moment. It should be noted that HTTP servers will respond with other HTTP error codes when responding to other error scenarios, as indicated in the HTTP specification.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
	O	C	DMS	7.8.9.3 If a serving endpoint cannot respond to an HTTP **GET** request with content byte data due to the device's lack of available network sockets, then the serving endpoint may refuse to create new TCP connections for answering content requests.	[14] [24]	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, rendering endpoints may 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.
	M	C	DMP	7.8.9.4 Rendering endpoints must not assume that the serving endpoint accepts more than one media transport HTTP connection at a time.	[24]	Procedures used in a rendering endpoint that require multiple simultaneous HTTP connections may not work for some serving endpoints. Examples of such procedures include: • 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.
	S	C	DMS	7.8.9.5 Serving endpoints should support more than one simultaneous HTTP media transport connections	[24]	It is a good practice for a serving endpoint to support multiple rendering endpoints simultaneously.
	S	C	DMP	7.8.9.6 If an HTTP server has not completed the transmission of an HTTP response and if the rendering endpoint wants to stop the current data flow to issue a new request, then the rendering endpoint's HTTP client should close the existing TCP connection and then create a new TCP connection for the new HTTP request.	[24]	Since rendering endpoints cannot assume that serving endpoints can support multiple HTTP connections, this guideline recommends that rendering endpoints use one TCP connection. Implementers should also consider this guideline in conjunction with guidelines 7.8.6.5 and 7.8.6.6, which deal with *scan operation* playback (a.k.a. trick-modes).

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.10 MT HTTP Header Tolerance	M	R	DMS, DMP	7.8.10.1 HTTP clients and servers must 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.	[19] [24]	This guideline addresses forward compatibility and allows for broader interoperability with implementations that employ transport-layer vendor-extensions by way of HTTP headers.
	M	R	DMS, DMP	7.8.10.2 Each HTTP header line (including the header's name and value but excluding the last carriage-return/line-feed sequence, CRLF) must not exceed 998 bytes.	[24] [25]	These guidelines limit the length of an HTTP header line according to section 2.2.1 of [25]. 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.
	M	R	DMS, DMP	7.8.10.3 If an HTTP header line (header's name and value but excluding the last CRLF) exceeds 998 bytes, then the HTTP header must span multiple lines. HTTP headers that span multiple lines must prefix the additional lines with at least one space (SP) or horizontal tab (HT) as described in section 4.2 of [24].		
7.8.11 MT HTTP Header Case-Sensitivity	M	R	DMS, DMP	7.8.11.1 Names of HTTP headers must be treated as case-insensitive tokens.	[19] [24]	This is normative according to the HTTP specification.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.12 MT Baseline Transport: HTTP Server	M	L	DMS	7.8.12.1 HTTP servers used for media transport purposes must be compliant with HTTP/1.1, which also requires HTTP/1.0 compliance.	[24]	HTTP servers need to support both HTTP/1.0 and HTTP/1.1 requests to ensure wide interoperability.
	S	R	DMS	7.8.12.2 HTTP/1.1 servers used for media transports should return HTTP version 1.1 in the response header, regardless of the version specified in the HTTP client's request.	[21]	The clarifying RFC ([21]) clarifies 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. It should be noted that *message format* refers to both the HTTP headers and HTTP response body. As described by [21], 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 may 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 [21] provides more details for interoperability between hosts with different HTTP versions.
	M	R	DMS	7.8.12.3 When responding to a request of version 1.0, the server must 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.	[21] [24]	
	M	R	DMS	7.8.12.4 HTTP servers must not report a higher version of HTTP than is actually supported by the implementation.	[24]	
	M	R	DMS, DMP	7.8.12.5 Interoperability between clients and servers that implement different versions of the HTTP protocol must follow the provisions and recommended actions defined in [21].	[21]	Reference [21] defines the significance of HTTP version numbers, the rules for interoperability between hosts with different version numbers, and rules for the actual version number that should be included when creating messages.
	S	R	DMS	7.8.12.6 HTTP servers should support persistent HTTP/1.1 connections and pipelined HTTP/1.1 requests.	[24]	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 may be used to facilitate seek operations.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.13 MT HTTP Header: **scmsFlag.dlna.org**	O	A	DMS, DMP	7.8.13.1 Serving endpoints may use the **scmsFlag.dlna.org** HTTP header, which indicates copyright assertion and copy status flags when transporting audio-only content. Servers that serve DLNA media content and non-DLNA media content may also use this flag for the latter. Rendering endpoints that encounter this HTTP header may implement behavior to enforce regional copyright provisions.	[24] [71]]	These guidelines make it possible to comply with regional legal requirements, such as in [71]. 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
	M	A	DMS	7.8.13.2 The notation of **scmsFlag.dlna.org** header field is defined as follows. • scmsFlag.dlna.org = "scmsFlag.dlna.org" ":" flagValue • flagValue = 00 | 01 | 10 | 11 The value of the **scmsFlag.dlna.org** header must 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		

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.14 MT HTTP Header: CONTENT-TYPE	M	L	DMS	7.8.14.1 HTTP servers used for media transport purposes must specify the **CONTENT-TYPE** HTTP header in the HTTP response header fields whenever it returns a Target Response.	[24]	It is imperative that HTTP servers specify the **CONTENT-TYPE** field to allow HTTP clients to know the MIME-TYPE for the content that is to be sent in the HTTP response message. It should be noted that MIME-TYPE values appear in the CONTENT-TYPE HTTP header.
	M	L	DMS	7.8.14.2 The MIME-TYPE values that appear in 7-5 must be used as values for **CONTENT-TYPE** when an HTTP message describes DLNA media contents.	N/A	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.
7.8.15 MT HTTP Header: **contentFeatures.dlna.org**	M	A	DMS	7.8.15.1 If an HTTP server receives an HTTP **GET** or **HEAD** request with the **getcontentFeatures.dlna.org** HTTP header, then the HTTP server must use the **contentFeatures.dlna.org** HTTP header if it responds with a Target Response. Note that HTTP servers may respond with **contentFeatures.dlna.org** for requests that do not have **getcontentFeatures.dlna.org**.	[24] [33]	As noted, this guideline permits a serving endpoint to use the **contentFeatures.dlna.org** in an HTTP response.
	O	A	DMP	7.8.15.2 HTTP clients of rendering endpoints may use the **getcontentFeatures.dlna.org** when issuing **GET** or **HEAD** requests.		These guidelines describe how a rendering endpoint can request an HTTP server to use the **contentFeatures.dlna.org** in the response.
	M	A	DMP	7.8.15.3 The notation of **getcontentFeatures.dlna.org** header field is defined as follows: • getcontentFeatures.dlna.org = “getcontentFeatures.dlna.org” “:” “1” The only value possible is “1”. Any other values sent will result in an error code response of 400 (Bad Request). Example: • getcontentFeatures.dlna.org: 1		

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
	M	A	DMS	7.8.15.4 The value of the **contentFeatures.dlna.org** HTTP header must be the same value as the fourth field of the content's res@protocolInfo value, as described in the 7.3.11, "MM DIDL-Lite protocolInfo values: 4th Field" guideline. The notation of **contentFeatures.dlna.org** header field for DLNA media transport is defined as follows; • contentFeatures.dlna.org = "contentFeatures.dlna.org" ":" 4th_field Please note that 4th_field is defined in guideline7.3.11.1. Note that this header may be used by serving endpoints for non-DLNA content.		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).

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.16 MT HTTP Header: dlna pragma-directive (**ifoFileURI.dlna.org**)	M	A	DMS	7.8.16.1 If a serving endpoint provides a dlna:res@ifoFileURI for a resource (see 7.3.38, "MM IFO File UR") and the HTTP server of the serving endpoint receives an HTTP **GET** or **HEAD** request with the Pragma-with-getIfoFileURI-pragma-directive in the HTTP request, then the HTTP server must provide the Pragma-with-ifoFileURI-pragma-directive if it responds with a Target Response. If a serving endpoint does not provide a dlna:res@ifoFileURI for a resource, then the HTTP server must not provide the Pragma-with-ifoFileURI-pragma-directive in the HTTP response.	[24] [33]	Section 14.32 of [24] 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 7.3.38, "MM IFO File UR", content profiled as MPEG_PS_NTSC or MPEG_PS_PAL may 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 rendering endpoint should be aware that the AV content does not have an IFO file. Implementors must be careful not to exceed the maximum 998 byte limit for HTTP header lines when using the Pragma-with-IfoFileURI-pragma-directive
	O	A	DMP	7.8.16.2 HTTP clients of rendering endpoints may use the Pragma-with-getIfoFileURI-pragma-directive when issuing **GET** or **HEAD** requests.		
	M	A	DMP	7.8.16.3 The notation of the **PRAGMA** header field with the getIfoFileURI-pragma-directive is defined as follows: • Pragma-with-getIfoFileURI-pragma-directive = "PRAGMA" ":" 1#pragma-directive • pragma-directive = getIfoFileURI-pragma-directive | "no-cache" | extension-pragma • getIfoFileURI-pragma-directive = "getIfoFileURI.dlna.org" extension-pragma = token ["=" (token | quoted-string)] Examples: • PRAGMA: getIfoFileURI.dlna.org • PRAGMA: getIfoFileURI.dlna.org, no-cache The HTTP header name is the "PRAGMA" string and the header value is the "getIfoFileURI.dlna.org" string. Both header name and value are treated as case insensitive strings. Note that the Pragma-with-getIfoFileURI-pragma-directive may be used by endpoints issuing an HTTP request on non-DLNA content.		

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
	M	A	DMS	7.8.16.4 The notation of the **PRAGMA** header field with the ifoFileURI-pragma-directive is defined as follows: • Pragma-with-ifoFileURI-pragma-directive = "PRAGMA" ":" 1#pragma-directive • pragma-directive = ifoFileURI-pragma-directive | "no-cache" | extension-pragma • ifoFileURI-pragma-directive = "ifoFileURI.dlna.org" "=" quoted-absolute-uri-string • quoted-absolute-uri-string = <same value as the corresponding dlna:res@ifoFileURI attribute value, as described in 7.3.38, "MM IFO File UR"> • extension-pragma = token ["=" (token | quoted-string)] Example: • PRAGMA: ifoFileURI.dlna.org= "http://192.168.0.1:8080/IFO_101.ifo" The HTTP header name is the "PRAGMA" string and the header value is the ifoFileURI-pragma-directive. Also the strings, " PRAGMA" and "ifoFileURI.dlna.org", are case insensitive, while the quoted-absolute-uri-string is a case sensitive URI value. A serving endpoint that provides the Pragma-with-ifoFileURI-pragma-directive with non-DLNA content must also provide an IFO file that conforms to the guidelines in 7.7.18.		.
7.8.17 MT HTTP **HEAD** Requests	M	R	DMS	7.8.17.1 HTTP servers (HTTP/1.1 and HTTP/1.0) must respond to HTTP **HEAD** requests, using the guidelines described below.	[24]	There are several interpretations to the format of **HEAD** responses. These guidelines provide consistent interpretation.
	M	C	DMS	7.8.17.2 A Target Response to a **HEAD** request must be composed of only the HTTP headers (of the equivalent **GET** response) and a zero-length response entity body.		
	M	C	DMS	7.8.17.3 Target Responses to **HEAD** requests that include the **CONTENT LENGTH** HTTP header field must supply the length of the response entity body for the equivalent **GET** Target Response.		This requirement properly allows HTTP clients to query an HTTP server for information on the size of content before receiving it.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
	M	C	DMS	7.8.17.4 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 must be omitted from the **HEAD** Target Response.		
	S	C	DMS	7.8.17.5 If an HTTP server (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.		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 may be different than for those of the **GET** request (e.g. server saturation, etc).
	M	L	DMS	7.8.17.6 The HTTP headers of a **HEAD** Target Response must include the **CONTENT-TYPE** HTTP header.		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, time-based seek operations, etc.) require a lot of computational cycles. For these reasons, these guidelines specify minimal expectations for **HEAD** responses while recommending the ideal expectations
	S	R	DMS	7.8.17.7 HTTP **HEAD** Target Responses should have the exact same HTTP headers as those in the equivalent HTTP **GET** Target Response.		
7.8.18 MT HTTP Header Parsing (Server)	M	R	DMS	HTTP servers must 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.	[24]	Incorrect HTTP header parsing has been the source of numerous compatibility issues during plugfest events.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.19 MT HTTP Header: **CONTENT-LENGTH**	M	C	DMS	7.8.19.1 If the **CONTENT-LENGTH** HTTP header is omitted from an HTTP **GET** response, then the HTTP server must do one of the following. The HTTP server will close the TCP connection immediately 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 must 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 must 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).	[24]	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 7.8.19.3. Likewise messages encoded with *Chunked Transfer Coding* will use the accurate value of "0" for the chunk-size.
	M	R	DMS	7.8.19.2 A response message with no entity body that uses *Chunked Transfer Coding* must use a chunk-size value of 0.	[24]	

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
	M	C	DMP	7.8.19.3 When operating under persistent connections (including pipelining), an HTTP/1.1 client detects 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.	[24]	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.
	S	C	DMS	7.8.19.4 If an HTTP server knows the byte length of a response body, then the HTTP server should use the **CONTENT-LENGTH** HTTP header in the HTTP Target Response.	[24]	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 streams (and some transcoded content) may 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.
	M	C	DMS	7.8.19.5 If the HTTP server does not know the byte length of the response entity body or if **CONTENT-LENGTH** cannot be used due to some exceptions listed in [24] section 4.4, then HTTP servers must omit the **CONTENT-LENGTH** HTTP header from **HEAD** and **GET** Target Responses.	[24]	
	M	C	DMS	7.8.19.6 If an HTTP server sends an HTTP **GET** Target Response with the **CONTENT-LENGTH** HTTP header, then the byte length of the response entity body must match the value of the **CONTENT-LENGTH** HTTP header.	[24]	
	M	R	DMS	7.8.19.7 If an HTTP server 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 must close the TCP connection when the end of the stream is reached.	[24]	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.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.20 MT Maximum Byte Size Transfers	M	L	DMS, DMP	7.8.20.1 HTTP clients and servers must 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 sections 14.35.1 and 14.16 of [24] and guideline 7.8.22.3) • instance-length (as defined in section 14.16 of [24]) • chunk-size (as defined in section 3.6.1 of [24])	[24]	The HTTP specification ([24]) 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.
	M	L	DMP	7.8.20.2 A rendering endpoint must 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.	[24]	
7.8.21 MT HTTP/1.0 Persistent Connections (Server)	M	L	DMS	7.8.21.1 If an HTTP server receives an HTTP/1.0 **GET** or **HEAD** request with the **CONNECTION: KEEPALIVE** header, then the HTTP server must ignore this header and close the TCP connection after the response is sent.	[19] [24]	Although HTTP/1.0 supports persistent connections by way of the KeepAlive token, there are interoperability issues because the methodology is an experimental extension.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.22 MT HTTP Header: **RANGE** (Server)	M	L	DMS	7.8.22.1 The HTTP server must support the **RANGE** header field defined in [24] for the HTTP **GET** and **HEAD** methods and follow the rules that are listed in the guidelines below:	[24]	HTTP servers 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**.
	O	C	DMS	7.8.22.2 If an HTTP server receives the **RANGE** HTTP header in a **HEAD** request, then the HTTP server may respond without the **CONTENT-RANGE** HTTP header.	[24]	HTTP servers are required to support the **HEAD** method. When an HTTP server gets a **HEAD** request with the **RANGE** option, the HTTP server may omit the **CONTENT-RANGE**.
	M	L	DMS, DMP	7.8.22.3 The notation of **RANGE** header field for DLNA media transport is a subset of the allowed syntax for [24], as stated below: • Range = "Range" ":" 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 Examples: Range: bytes=1539686400- Range: bytes=1539686400-1540210688	[24]	This guideline simplifies the implementation of HTTP servers by requiring only a subset of the allowed **RANGE** syntax afforded by [24]. 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 must be used within the **RANGE** header field.
	M	L	DMS	7.8.22.4 If the HTTP server returns data including the requested range (Target Response), it must specify the **CONTENT-RANGE** header field. Furthermore, the HTTP response code must be: 206 (Partial Content). Example of **CONTENT-RANGE** header. Content-Range: bytes 1539686400-1540210688/9238118400	[24]	These guidelines obligate an HTTP server to respond with either a 206 response code (for a request that can be honored) or a 416 response code (for requests that cannot be honored). These guidelines simplify HTTP client implementations as they limit the HTTP server to more predictable behavior. Section 14.16 in [24] has examples on this guideline usage.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
	M	C	DMS	7.8.22.5 If the HTTP server uses the **CONTENT-RANGE** HTTP header, then the provided values must be accurate with respect to the entity response body. Specifically, The value indicating the first-byte-pos must properly match the first byte of the response entity body. The first-byte-pos in the response must match the first-byte-pos in the request message. The value indicating the last-byte-pos must properly match the last byte of the response entity body. The value indicating the instance-length must indicate the length of the entire content binary or asterisk (*) if unknown.	[24]	
	M	C	DMS	7.8.22.6 If the requested range is not valid for the resource with a URI specified in the HTTP request, the HTTP server must respond with the HTTP response code of: 416 (Requested Range Not Satisfiable). When encountering syntax errors with the **RANGE** HTTP header, the HTTP server must use the HTTP response code 400 (Bad Request).	[24]	
	M	L	DMS	7.8.22.7 If any requested range for the specified URI can never be processed/satisfied by the HTTP server, for example, in the case of real-time transcoding or live contents, the HTTP server must respond with 406 (Not Acceptable).	[24]	
	S	A	DMS	7.8.22.8 If an HTTP server can support HTTP requests with a specified **RANGE** for a particular URI, then the HTTP server should support persistent connections (HTTP/1.1) for that URI.	[24]	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.
	S	A	DMS	7.8.22.9 HTTP servers should accept and honor an HTTP/1.0 **GET** or **HEAD** requests with the **RANGE** header field.	[24]	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**.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.23 MT HTTP Chunked Transfer Coding	O	R	DMS	7.8.23.1 HTTP servers may use *Chunked Transfer Coding* in response to HTTP/1.1 **GET** requests.	[24]	*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.
	M	R	DMS	7.8.23.2 HTTP servers must not use *Chunked Transfer Coding* in response to HTTP/1.0 **GET** requests.	[24]	
7.8.24 MT HTTP Time-Based Seek (Server)	S	A	DMS	7.8.24.1 HTTP servers should support the **TimeSeekRange.dlna.org** HTTP header field, which is defined by DLNA, for the transport of DLNA audio/visual and audio media.	[24]	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. It should be noted that HTTP clients may also use seek operations to implement a forward/backward variable-speed playback capability.
	O	C	DMS	7.8.24.2 If an HTTP server receives the **TimeSeekRange.dlna.org** HTTP header in a **HEAD** request, then the HTTP server may respond without the **TimeSeekRange.dlna.org** HTTP header.	[24]	HTTP servers are required to support the **HEAD** method. When an HTTP server gets a **HEAD** request with this option, the HTTP server may omit it from the response.
	O	A	DMS	7.8.24.3 An HTTP server that serves DLNA content and non-DLNA content may also support the **TimeSeekRange.dlna.org** HTTP header field for the latter type of content.	[24]	This guideline permits DLNA implementations to use this HTTP header for content that does not conform to a DLNA media format profile.
	M	A	DMS, DMP	7.8.24.4 The notation of the **TimeSeekRange.dlna.org** header field is defined as follows. • TimeSeekRange.dlna.org = "TimeSeekRange.dlna.org" ":" range-specifier • range-specifier = npt-range [bytes-range] • npt-range = “npt=“ npt-time "-" [npt-time] [instance-duration] • instance-duration = “/” (npt-time | “*”) • npt-time = npt-sec | npt-hhmmss • npt-sec = 1*DIGIT ["." 1*3DIGIT] • npt-hhmmss = npt-hh ":" npt-mm ":" npt-ss ["." 1*3DIGIT]	[24]	The field value specifies the requested range of a resource in absolute time. The range is specified by the start point and the endpoint. If the endpoint is omitted, it means the end of the resource is specified. This is similar to byte-range-specifier used for the **RANGE** header field.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
				• npt-hh = 1*DIGIT ; any positive number • npt-mm = 1*2DIGIT ; 0-59 • npt-ss = 1*2DIGIT ; 0-59 • • byte-range = “bytes=” first-byte-pos "-" last-byte-pos instance-length • first-byte-pos = 1*DIGIT • last-byte-pos = 1*DIGIT • instance-length = “/” (1*DIGIT | “*:”) The npt-range specifies the range in normal playing time and 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 7.8.24.6 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 7.8.23.9 for more information. The instance-length specifies the byte length of an entire content binary and is mandatory in the response in conjunction with the byte-range. The instance-length is prohibited in the request. The asterisk "*" character means that the instance-length is unknown at the time when the response was generated. Refer to 7.8.24.6. 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.		

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
	M	A	DMS	7.8.24.5 If an HTTP server returns data (Target Response) including the requested time-range, it must return the content bytes for a time-range that starts at or before the requested start point and ends at or after the requested endpoint. The following exceptions are allowed. Serving endpoint may ignore the requested endpoint and return the range data up to the end of the content data. • Serving 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/* If the requesting endpoint specifies a time beyond the end of the content data, then the serving endpoint must return the range data to the end of the content data.	[24]	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.
	M	A	DMS	7.8.24.6 If an HTTP server supports both byte range-based seek and time-based-seek for a resource, it must specify a byte-range value as well as an npt-range value in the HTTP Target Response to the HTTP request with TimeSeekRange.dlna.org header field. The npt-range and byte-range must 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/*		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, one should use multiple HTTP **GET** requests with the **RANGE** header field to retrieve subsequent content data. 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 time-based seek and byte/range-based seek) to specify both the time-range and a byte-range in the HTTP response's headers.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
	S	A	DMS	7.8.24.7 If an HTTP server 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).		This guideline recommends that stream segments returned by servers should start, if possible, with a recognizable decoding entry point, such as the headers in a group of pictures (GOP).
	M	A	DMS	7.8.24.8 If an HTTP server returns data (Target Response) from a **GET** request with time range, it must use the **TimeSeekRange.dlna.org** header field to indicate time-range of the content data that is returned in the HTTP response. Furthermore, the HTTP response code must be 200 (OK). The npt-range must 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/*	[24]	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.
	M	A	DMS	7.8.24.9 If the requested time range is *not valid* for the resource with URI specified in the HTTP **GET** request, then the HTTP server must 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. If the requested time range is not syntactically correct, then the HTTP server must return the HTTP response error code of 400 (Bad Request). If any HTTP request with **TimeSeekRange.dlna.org** for the specified URI can never be processed/satisfied by the HTTP server (for example, in the case of real-time transcoding or live contents), then the HTTP server must respond with 406 (Not Acceptable).	[24]	

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.25 MT Seek Behavior for Stored Content	M	A	DMS, DMP	7.8.25.1 The npt value of 0 for the **TimeSeekRange.dlna.org** header field and the byte value of 0 must refer to the beginning of a content binary.	[24]	This guideline clarifies the behavior of time-based-seek and byte-based-seek operations for stored contents. If the npt value of 0 is specified in the first position of the **TimeSeekRange.dlna.org** header filed of HTTP **GET** request, the serving endpoint returns content data that starts at the beginning of a content binary as well as HTTP **GET** requests without the **TimeSeekRange.dlna.org** header field. Serving endpoints which cannot satisfy the seek behavior for a URL (such as live contents) must not indicate the seek capabilities in the op-param of the protocolInfo's 4th field. (See 7.3.11.4) or in the **contentFeatures.dlna.org** HTTP header (see 7.8.15)
	M	C	DMS, DMP	7.8.25.2 The byte position of 0 for the **RANGE** header field must refer to the beginning of a content binary.		
	M	A	DMS	7.8.25.3 If a serving endpoint supports the **TimeSeekRange.dlna.org** header field for the specified URI, it must process any requested range from the beginning to the end of the content binary.		
	M	C	DMS	7.8.25.4 If a serving endpoint supports the **RANGE** header field for the specified URI, it must process any requested range from the beginning to the end of the content binary.		

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.26 MT HTTP Server-side Trick Mode	O	A	DMS	7.8.26.1 The HTTP server may support the **PlaySpeed.dlna.org** HTTP header field, for the transport of audio/visual and audio media class for variable speed playback. It should be noted that this HTTP header may be used by DLNA servers for content conforming to DLNA media format profiles and content that does not conform to DLNA media format profiles.	[24]	This 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.
	O	C	DMS	7.8.26.2 If an HTTP server receives the **PlaySpeed.dlna.org** HTTP header in a **HEAD** request, then the HTTP server may respond without the **PlaySpeed.dlna.org** HTTP header.	[24]	HTTP servers are required to support the **HEAD** method. When an HTTP server gets a **HEAD** request with this option, the HTTP server may omit it from the response.
	M	A	DMS, DMP	7.8.26.3 If the HTTP server or client uses the **PlaySpeed.dlna.org** HTTP header, then the HTTP client or server must use the following syntax for the HTTP header and its value. • PlaySpeed.dlna.org = "PlaySpeed.dlna.org" ":" play-speed-specifier • play-speed-specifier = "speed" "=" TransportPlaySpeed (* The notation of TransportPlaySpeed is defined by UPnP AV Transport service type.) 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 must use the HTTP response code 400 (Bad Request).	[24]	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.).
	M	A	DMS	7.8.26.4 If the HTTP server returns data (Target Response) for scaled content to be decoded for a trick-mode/variable-play-speed, the HTTP response message must use the **PlaySpeed.dlna.org** HTTP header field to indicate the play speed of the scaled content. Furthermore, the HTTP response code must be 200 (OK).	[24]	This guideline requires the HTTP server to indicate if content bytes in the HTTP response represent content that has been time-scaled.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
	M	A	DMS	7.8.26.5 If any requested **PlaySpeed.dlna.org** for the specified URI can never be processed/satisfied by the HTTP server, for example, in the case of real-time transcoding or live contents, the HTTP server must respond with 406 (Not Acceptable). If the requested play speed is not valid for the resource with URI specified in the HTTP **GET** request, then the HTTP server must respond with the HTTP response error code 406 (Not Acceptable). Interpretation of not valid includes the following types of errors: The requested play-speed range is not supported for the content.	[24]	This guideline specifies the error code to be used in scenarios where the HTTP server cannot accommodate a request to time-scale content.
	M	A	DMS	7.8.26.6 The scaled data (returned by the HTTP server as a Target Response) must be compliant to the media format profile indicated in the corresponding <res> element, obtained from a UPnP AV ContentDirectory service implementation.	[24] [33]	This guideline obligates a serving endpoint to return content bytes that are conformant to the characteristics described for that content by the associated UPnP AV ContentDirectory service.
7.8.27 MT Combined **RANGE**, Time-based Seek, and Play-Speed HTTP Requests	M	A	DMS, DMP	7.8.27.1 If an HTTP server receives an HTTP **GET** request with both the **PlaySpeed.dlna.org** and the **TimeSeekRange.dlna.org** header fields, the HTTP server must understand that time-scaling is requested for the specified time range If the HTTP server can never process either or both time-scaling and time-seek, the error code for time-scaling, 406(Not Acceptable) must be returned.	[24]	This guideline covers what a serving endpoint needs to do if it receives an HTTP request for both time-based seek and play speed. This guideline also infers how a rendering endpoint can expect a serving endpoint to behave.
	M	A	DMS, DMP	7.8.27.2 If an HTTP server receives an HTTP **GET** request with **RANGE** and (**TimeSeekRange.dlna.org** or **PlaySpeed.dlna.org**) header fields, then the **RANGE** header field must take the highest precedence and the server should ignore the other time-seek and play-speed fields.	[24]	This guideline covers what a serving 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 rendering endpoint can expect a serving endpoint behaves.

Table 7-8 HTTP Media Transport Guidelines (Continued)

Name	M/S/	A/C/F	Dev Class	Description	Ref #	Comments
7.8.28 MT Baseline Transport: HTTP Client	M	L	DMP	7.8.28.1 HTTP clients used for media transport purposes must implement HTTP/1.0, HTTP/1.1, or both.	[24]	HTTP clients are restricted to HTTP versions that HTTP servers are prepared to support.
	M	R	DMP	7.8.28.2 HTTP clients must not report a higher version of HTTP than is actually supported by the implementation.	[24]	For example an HTTP client that does not support *Chunked Transfer Coding* responses must never issue an HTTP/1.1 **GET** request.
	M	R	DMP	7.8.28.3 HTTP/1.1 clients must be able to process HTTP/1.1 *Chunked Transfer Coding* responses.	[24]	When making HTTP/1.1 requests, it is important that HTTP clients properly handle responses encoded with *Chunked Transfer Coding*.
	M	R	DMP	7.8.28.4 HTTP/1.1 clients must be prepared to properly handle an HTTP response code of 100 (Continue Response) from HTTP servers, even when not expected.	[24]	A 100 (Continue Response) can be generated by the server regardless of whether or not the client issued an HTTP request encoded with *Chunked Transfer Coding*.
7.8.29 MT HTTP Header: **RANGE** (Client)	M	L	DMP	7.8.29.1 HTTP clients must not use multiple range specifiers nor use *suffix-byte-range-spec* (as defined in [24]) in HTTP requests.	[24]	This guideline simplifies the implementation of HTTP servers by not requiring support of multiple ranges or *suffix-byte-range-spec*.
7.8.30 MT HTTP Persistent Connection Usage for Clients	S	R	DMP	7.8.30.1 HTTP/1.1 clients should use HTTP/1.1 persistent connections.	[24]	Implementing this guideline reduces the setup/teardown load on serving endpoints. Furthermore, serving endpoints will be able to reserve the allocated socket for the requesting client. Clients that do not use HTTP/1.1 persistent connections may encounter a scenario where a serving endpoint does not answer the subsequent requests because it has run out of sockets.
	M	L	DMP	7.8.30.2 HTTP/1.0 clients must not use the **CONNECTION: KEEPALIVE** token for HTTP/1.0 transactions.	[19]	See the 7.8.21, "MT HTTP/1.0 Persistent Connections (Server)"guideline for more information.
7.8.31 MT HTTP Inactivity Timeout	S	A	DMP	7.8.31.1 HTTP clients should close persistent (HTTP/1.1) connections after completing all outstanding HTTP transactions and within 30 seconds of inactivity has passed.	[24]	This ensures that sockets do not remain consumed after a content transfer has successfully completed.
7.8.32 MT HTTP Header Parsing (Client)	M	R	DMS	7.8.32.1 HTTP clients must 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.	[24]	Incorrect HTTP header parsing has been the source of numerous compatibility issues during plugfest events.

APPENDIX A (INFORMATIVE)

A

HOME NETWORK INFRASTRUCTURE RECOMMENDATIONS

Home network infrastructure devices are outside the scope of the DLNA Home Networked Device Interoperability Guideline. However, since DLNA devices interact with each other on a home network, that network and its infrastructure greatly influence the user experience. Home network infrastructure devices that abide by the recommendations in this section will contribute to and facilitate interoperability and a good user experience with DLNA devices. Although this document lists recommendations, a home network infrastructure device may 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.1 NETWORK INFRASTRUCTURE DEVICE FUNCTIONS

The recommendations in Table A-2 refer to different types of home network infrastructure functionality. Home network infrastructure devices may be single-function devices, such as a switch, or they may be combination devices that implement multiple functions such as a wireless access point that also provides Ethernet ports with bridging between wired and wireless interfaces. The device functions referenced in the recommendations are defined in Table A-1.

Table A-1 Home Network Infrastructure Device Functions

Device Function	Descriptions
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.
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.
Router	Routers pass traffic between two or more IP subnets and, within a single subnet, perform address resolution of IP addresses. Routers may 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.2 NETWORK INFRASTRUCTURE RECOMMENDATIONS

Table A-2 Home Network Infrastructure Recommendations

Name	M/S/O	A/C/F/L/R	Description	Ref #	Comments
A.2.1 NCI Ethernet: Base	S	R	A.2.1.1 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.	[1]	
A.2.2 NCI Ethernet: Cabling	S	R	A.2.2.1 If Ethernet is supported, any supplied network cabling should have a rating of Category 5e or better.	[2]	
A.2.3 NCI Ethernet: Gigabit	O	R	A.2.3.1 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.	[1]	

Table A-2 Home Network Infrastructure Recommendations (Continued)

Name	M/S/O	A/C/F/L/R	Description	Ref #	Comments
A.2.4 NCI Ethernet: QoS Tolerance	S	R	A.2.4.1 If Ethernet is supported, tagged packets should be tolerated. Tagged packets are Ethernet packets that include priority tags conformant with [1], section 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.	[1] [67]	
A.2.5 NCI IGD: LAN-Side IP Stack	S	R	A.2.5.1 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 [17] .	[11] [12] [13] [14] [15] [16] [17]	
A.2.6 NCI IGD: LAN-Side DHCP	S	R	A.2.6.1 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.	[20]	
A.2.7 NCI IGD: LAN-Side DNS	S	R	A.2.7.1 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.	[70]	
A.2.8 NCI IGD: NAT	S	R	A.2.8.1 IGDs should support NAT functionality between their LAN-side and WAN-side interfaces.	[69]	
A.2.9 NCI IGD: Upgradeability	S		A.2.9.1 IGDs should be firmware updatable by the end user.		

Table A-2 Home Network Infrastructure Recommendations (Continued)

Name	M/S/O	A/C/F/L/R	Description	Ref #	Comments
A.2.10 NCI AP: Connectivity	S	R	A.2.10.1 APs should support 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 [NCI Ethernet:] labeled requirements in this table with bridging between the Ethernet and 802.11 segments.	[3] [5]	Note that 802.11g also includes support for 802.11b.
A.2.11 NCI AP: Wi-Fi Conformance	S	R	A.2.11.1 APs should conform to Wi-Fi test plan requirements at the time the product is offered to the market.	[7] [8] [9] [10]	Wi-Fi interoperability requirements are increasing with time as new capabilities and features are specified by IEEE 802.11. Examples of these features which are not tested by Wi-Fi at this writing, but which are expected to be in the near future, include advanced link security and link-level QoS. When these capabilities are added to the Wi-Fi certification test plans, wireless implementations should conform to them.
A.2.12 NCI AP: Upgradeability	S		A.2.12.1 APs should be firmware updatable by the end user.		
A.2.13 NCI Bridge: Addressability	S		A.2.13.1 All bridges should be IP addressable and have a unique IP address (layer 3) so they may be managed through IP or higher layer protocols.		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 should be IP addressable so that the specific management solution can be invoked over the network.

APPENDIX B (INFORMATIVE)

B

TUNER REPRESENTATION

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 Appendix 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.1 TUNER OBJECTS

The Tuner is represented as a CDS container (object.container) object. If a serving endpoint has two or more identical tuners (for example a device with two NTSC analog tuners), each tuner may be represented as a separate container object, or these tuners may 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 may be uniquely selected. A 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 Tuner Container must have a <dlna:containerType> property with a value of “Tuner_1_0” to allow Control Points to differentiate them from other Container types.

B.2 CHANNEL OBJECTS

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 Tuner Container should contain a videoBroadcast or audioBroadcast item to represent each tunable (or selectable) channel. A 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.

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 7.3.30).

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 7.3.34.1).

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 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 7.3.35.1).

CHANNEL TITLE

The Channel Title is represented in the dc:title property, which all CDS items must 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 7.3.33.1).

B.3 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 serving endpoint 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 serving endpoint 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 serving endpoint should refuse such connection requests that cannot be accommodated and return an HTTP error code of 503 (Service Unavailable). A separate HTTP connection must 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 serving endpoint 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 serving endpoint 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 serving endpoint to stream the Channel content, and later the serving endpoint changes the "current" channel, the serving endpoint 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 serving endpoint. 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.4 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 Tuner container utilizes channel numbers based on broadcast channels while the FM 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 7.2.29 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 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_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_PN=MPEG_PS_NTSC">
      http://192.168.0.20:58849/Tuner1/ch4.mpg
    </res>
  </item>
</container>
<!-- FM Radio 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_PN=LPCM">
      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_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_PN=LPCM">
      http://192.168.0.20:58849/Tuner2/ch3.L16
```

```
        </res>
      </item>
    </container>
  </container>
</DIDL-Lite>
```

APPENDIX B (INFORMATIVE)

APPENDIX C (INFORMATIVE)

C

UPNP DEVICES WITH MULTIPLE NETWORK INTERFACES

C.1 REPRESENTATION AT THE UPNP DEVICE LEVEL

This appendix 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: 7.2.6, "DDC UPnP HTTP/1.0 Rules" and 7.3.19, "MM DIDL-Lite Content: Multiple Points of Accessibility". This appendix 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 7.2.26.3, control points also obtain the correct IP address for each logical UPnP device.

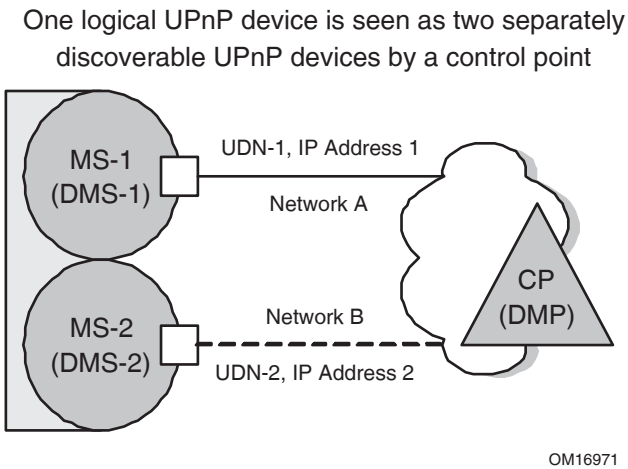


Figure C.1. UPnP Device Representation

An important observation is that Network A and Network B may 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 may not be able to identify the discoverable UPnP devices as part of a common logical UPnP device, additional meta-information may 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.

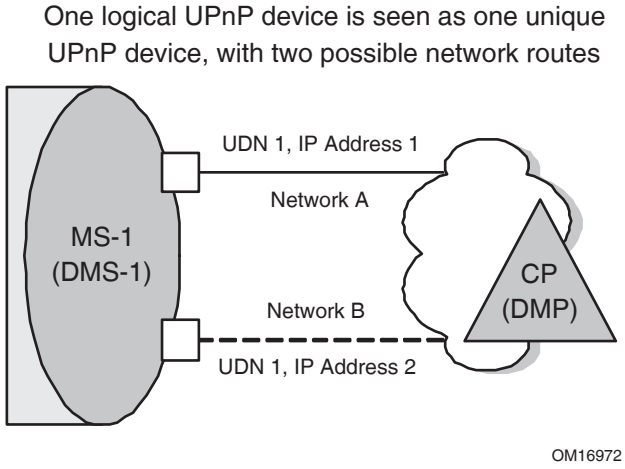


Figure C.2. UPnP Device on Multiple Networks

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 may 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 must 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 7.2.26.3 and 7.2.26.4 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.

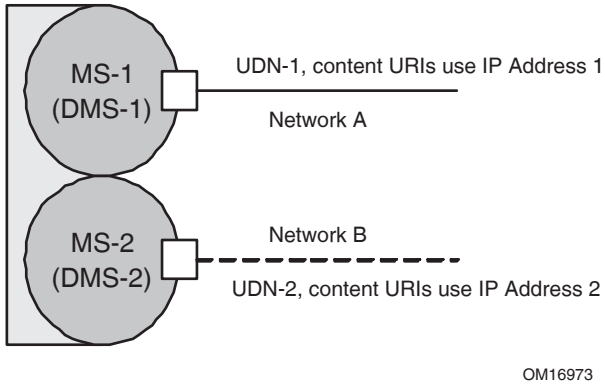


Figure C.3. Representation at the CDS Level

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 7.3.19.1. 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 may 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.

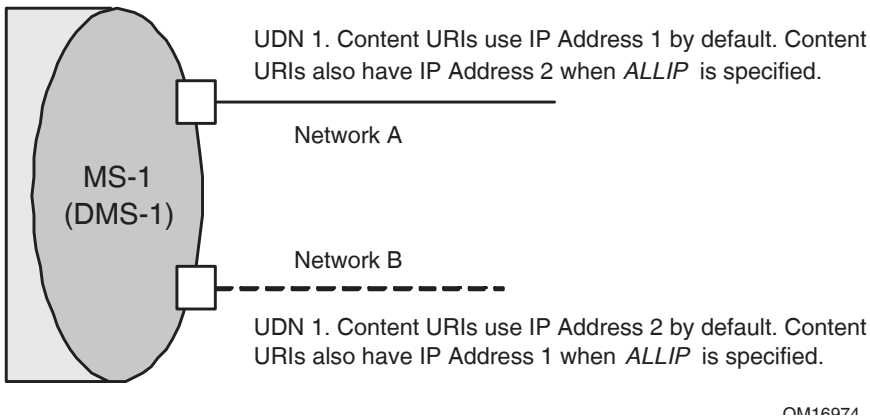


Figure C.4. Content URIs over Multiple Networks

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 7.3.19.1, which requires DMS-1 to never report URI values that have a Network B addresscs, 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 section, 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 7.3.19.1. 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 must 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 7.3.19.3 and 7.3.19.4 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 may 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.

APPENDIX C (INFORMATIVE)