



Allegro

RDMC 101

Users Manual **Version 1.20**

December 2011

Allegro Software Development Corporation

1740 Massachusetts Avenue

Boxborough, MA 01719

Telephone: 978 264-6600 Fax: 978 266-2839

www.allegrosoft.com

COPYRIGHTS

This manual is © Copyright 1996-2011 by Allegro Software Development Corporation. All rights reserved. RomPager®, RomXML®, RomPlug® and RomXOAP® are trademarks registered in the U.S. Patent and Trademark Office to Allegro Software Development Corporation. RomCLI, RomDNS, RomMailer, RomPOP, RomRadius, RomSShell, RomTime, RomWebClient, and Allegro Media Server are trademarks of Allegro Software Development Corporation. The RomPager and RomMailer products described in this manual are protected by U.S. Patent No. 6,782,427. UPnP™ is a certification mark of the UPnP Implementers Corporation. The name of other companies and products mentioned herein are the trademarks of their respective owners.

TABLE OF CONTENTS

Copyrights	ii
Table of Contents	iii
CHAPTER 1	
Introduction	1
RomPager Product Family	1
CHAPTER 2	
UPNP / DLNA Overview	4
UPnP™/DLNA® Overview	4
<i>Evolution of DLNA</i>	4
<i>DLNA Device Model</i>	5
<i>DLNA Device Categories and Classes</i>	6
DLNA Architecture	12
<i>DLNA Media Formats</i>	12
<i>DLNA Device Discovery, Description, Control, Eventing and Media Management</i>	13
<i>Device Discovery</i>	13
<i>Description</i>	13
<i>Control</i>	13
<i>Eventing</i>	13
<i>Presentation</i>	14
<i>Media Management</i>	14
<i>Content Directory</i>	14
<i>Connection Manager</i>	15
<i>AV Transport</i>	15
<i>Rendering Control</i>	15
<i>DLNA Media Transport</i>	16
<i>Network Stack</i>	16
<i>Network Connectivity</i>	16
Link Protection and Digital Rights Management (DRM)	16
Certification	16
CHAPTER 3	
RDMC Getting Started	18
RDMC 101 - Getting Started	18
RDMC 101 - Getting to Know the RDMC 101	18
Applicable Documents	20
RDMC 101 Requirements	20
Connecting the RDMC 101 to a Network	21

CHAPTER 4

Controlling DLNA Devices

22

Interacting with the RDMC 101	22
RDMC 101 HTTP Interface	23
DMS to DMR	25
<i>Selecting a Server</i>	26
<i>Selecting a Renderer</i>	26
<i>Controlling a Renderer</i>	26
DMS to DMS	27
DMS to DMPr	28
Preferences	30
<i>Seek Command Preference</i>	31
<i>DLNA Playlist File Handling</i>	31
<i>Play Behavior</i>	31
<i>Friendly Name</i>	31
<i>Check for Updates</i>	31
<i>Use Factory Preferences</i>	31
RDMC 101 Telnet Interface	33
browse	34
census	35
devicecaps	36
download	37
ff	39
find	40
findrenderers	41
findservers	42
formats	43
fr	46
help	47
metadata	49
mute	51
name	52
next	53
nexttrack	54
pause	55
play	56
playall	59
playmode	61
playnext	62
previous	64
previoustrack	66
quit	69
record	70
recordquality	71
resume	72
seek	74
select	75
show	76

status	78
stop	80
subscribe	82
transportactions	84
unsubscribe	85
up	87
upload	90
volume	91

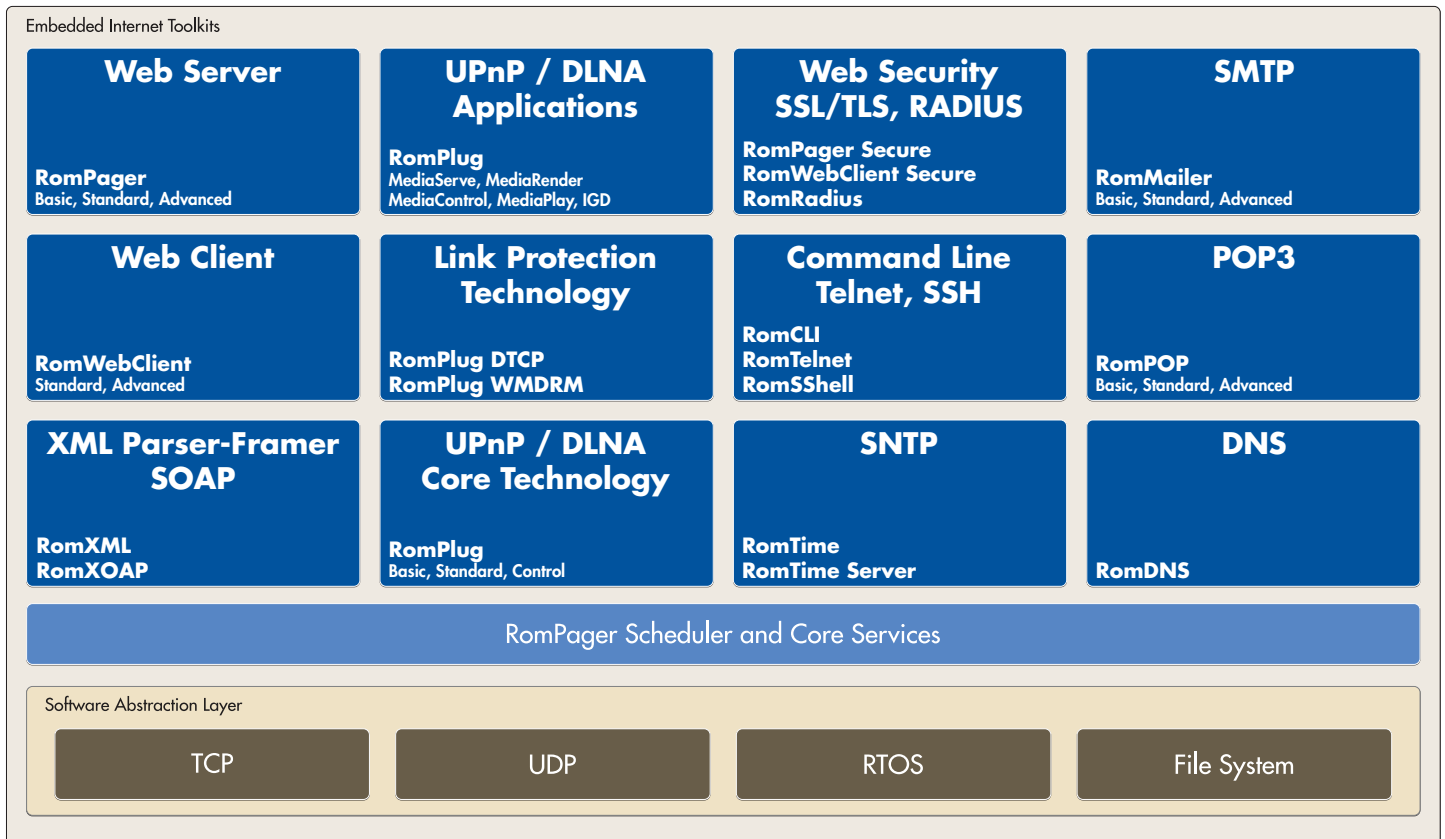
CHAPTER 1

INTRODUCTION

RomPager Product Family

The RomPager® family of products bring Web, email and other Internet services to embedded devices. The RomPager toolkits allow virtually any device to use standard Internet applications. A device can use Internet standard protocols to serve pages, images or applets; retrieve files from Web servers; and send or receive email with attachments; easily allowing Internet-based man-machine interaction. The RomPager products also allow embedded device designers to easily develop machine-to-machine systems using standard Internet techniques such as Web Services or SOAP.

The RomPager product family is provided in ANSI-C source code and has been ported to all major processor/ OS platforms and is delivered with interface files for the leading OS environments. The RomPager products use a common Software Abstraction Layer to provide an interface with any RTOS, TCP/IP and file system environment. With a typical OS, the RomPager product family shares a single task/thread in the device environment using a common light-weight scheduler to support multiple simultaneous HTTP and other protocol requests. In fact, the Allegro products can run in devices without an OS, by running off the idle loop. All of the protocol products will run separately or in combination with the others. Sophisticated compiler option flags allow maximum code-sharing to provide the smallest possible code footprint.



There are three RomPager Web Server toolkits that provide a range of embedded Web server capabilities. The **RomPager Basic** toolkit is an HTTP 1.0/1.1 Web server with CGI-style user exit support and optional file system support. It uses from 7Kb to 12Kb of ROM code and provides a small, powerful server for low-end devices. The **RomPager Standard** toolkit includes all the capabilities of the RomPager Basic toolkit and adds the PageLoader web object compiler. PageLoader imports Web pages (prepared with any Web page layout program), applets and graphics such as GIF, JPG and PNG and creates a compressed Web object library for the device. The **RomPager Advanced** toolkit includes all the capabilities of the RomPager Standard toolkit and provides additional HTTP 1.0/1.1 features, an internal security database and the PageBuilder compiler. The PageBuilder compiler provides the features of PageLoader and a comprehensive Web Application development environment including full support for HTML (2.0, 3.2, and 4.0), XHTML, Javascript, object compression, application compression and support for internationalization with dynamic phrase dictionaries.

Optional packages for the RomPager Advanced Web server include **SoftPages**, **Remote Host** and **Graphlets**. SoftPages adds a runtime HTML parser to RomPager and allows the device vendor to make runtime source changes to HTML pages. Remote Host provides integrated HTTP proxy services to support redirection of HTTP requests from the RomPager Web server to another Web server for retrieving objects too large to store in the embedded device. The Graphlets toolkit is a series of Java applets that provide graphic control indicators for the embedded device. The applets include line charts, bar charts, progress bars and dial indicators.

RomWebClient™ Standard is an HTTP 1.0/1.1 client that provides embedded devices the ability to retrieve and store objects from remoteWeb servers using the HTTP protocol. Objects can be in any format and can be used in a memory buffer or stored in the optional file system. **RomWebClient Advanced** adds caching, cookies and pipelining capabilities. The Web clients interoperate with any standard Web server or with other embedded devices that have embedded Web servers.

The **RomPager Secure** and **RomWebClient Secure** toolkits provide SSL 3.0 and TLS 1.0, 1.1 secure server and client sessions. The encryption protocols interoperate with any secure browser or server and include RSA, RC4 DES, 3DES, SHA and AES algorithms. The RomWebPager and RomWebClient secure toolkits offer standalone security capabilities or are available as integrated options for the RomPager server or RomWebClient toolkits.

RomPlug® Basic is a toolkit for implementing the Device Discovery and Description sections of the UPnP™ architecture. The **RomPlug Advanced** Device toolkit adds support for the Control and Eventing sections of the UPnP architecture to build certified UPnP devices. The **RomPlug Control** toolkit provides the ability to build fully compliant UPnP architecture Control Points. Optional application toolkits provide support for **UPnP IGD**, **Media Renderer**, **Media Server**, **Media Player** and **Media Controller** devices. The **RomPlug DTCP** and **RomPlug WMDRM** toolkits allow engineering teams to easily integrate link protection into state of the art UPnP and DLNA enabled consumer electronics and mobile devices.

RomCLI™ is a toolkit that may be used to build Command Line Interfaces similar to Cisco IOS-based products. The RomCLI toolkit includes an offline compiler (CliBuilder) for preparing command definitions and also includes a built-in Telnet server as well as serial support. The **RomSShell™** toolkit provides a SSH Version 2 server that may be used by itself or as an integrated front end to RomCLI.

RomMailer™ Basic is a Simple Mail Transfer Protocol (SMTP) client that enables embedded devices to send Internet email text messages. **RomMailer Standard** adds support for attachment files using MIME and UUENCODED formats. **RomMailer Advanced** provides Delivery Status Notification and Message Delivery Notification support. When RomMailer is used with the RomPager Advanced Server, messages can also be HTML Mail with embedded graphics and dynamic insertion of variables into the message text.

RomPOP™ Basic is a toolkit for building a Post Office Protocol (POP3) client so that embedded devices can receive Internet email text messages. **RomPOP Standard** adds support for attachment files using MIME and UUENCODED formats. **RomPOP Advanced** provides Delivery Status Notification and Message Delivery

Notification support.

RomDNS™ is a Domain Name Services client that provides embedded devices the ability to perform lookups of a variety of DNS records. It may be used to simplify configuration for RomMailer and RomPOP, or to provide server addresses for RomWebClient.

RomTime™ is a Network Time Protocol (NTP) client that provides embedded devices the ability to receive time services from a network time server, allowing devices to avoid requiring the user to enter the time manually.

RomTime Server is an NTP server designed specifically for use in embedded applications.

The **RomXML®** toolkit is a small eXtensible Markup Language (XML) implementation that enables your embedded device to send (frame) and receive (parse) XML documents. Using XML in your embedded designs provides for free-format interchange of data and is widely accepted in the device management, remote sensing and enterprise IT communities. Allegro's RomXML has been designed from the ground up for use in embedded devices that often have limited resources. Written in ANSI-C, the toolkit offers built in capabilities to convert internal data between C language structures and XML documents. The **RomXOAP®** toolkit builds upon the capabilities of RomXML and offers design engineers a comprehensive solution for creating connectivity between embedded designs and enterprise IT environments utilizing standards based SOAP technology. Available as standalone toolkits or tightly integrated with the other RomPager family of products, RomXML and RomXoap provide the foundation for enabling embedded devices with XML, SOAP and Web Services capabilities.

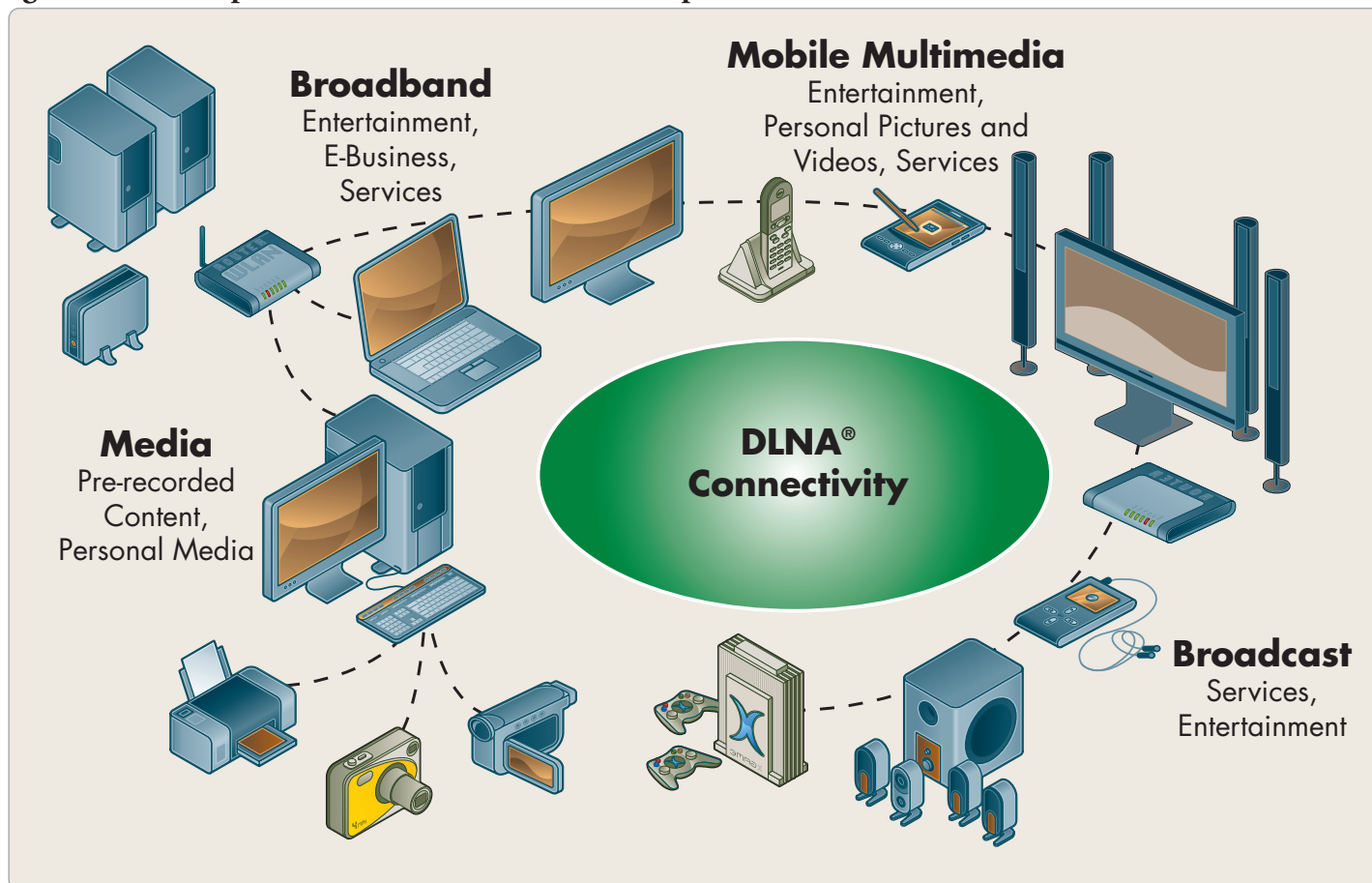
CHAPTER 2

UPNP / DLNA OVERVIEW

UPnP™/DLNA® Overview

Consumers have embraced digital technologies and given rise to the digital home. Over the past few years consumers have started to acquire, enjoy and manage an ever increasing amount of digital content. A large library of digital music, various photos from digital cameras, cell phones, home videos and more recently premium video content downloaded from the Internet are just a few examples. The scale, options and size of digital content will only grow over the coming years. Along with the shift to digital content, consumers want an environment that allows all their devices in the home to easily work together regardless of vendor. This vision of the digital home integrates Personal Computers (PC), Consumer Electronics (CE) and Mobile Devices (MD) with a seamless interoperable network. It is also one that requires a common set of industry design guidelines to allow companies to build pervasive interoperability.

Figure 2.1 - Interoperable DLNA home network example



Evolution of DLNA

The Digital Living Network Alliance (DLNA) was formed to address the need for a common set of industry design

guidelines. The DLNA Home Networked Device Interoperability Guidelines were created by a unique cross-industry effort that combined the efforts of over 250 Consumer Electronics, PC and Mobile Device companies from around the world who worked together with the aim of achieving the world's first substantial approach to true interoperability between personal computers, consumer electronics, and mobile 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 ability of DLNA to deliver workable interoperability guidelines in less than 12 months is largely due to the pioneering efforts of the Universal Plug and Play Forum (UPnP Forum - www.upnp.org). The mission of the Forum is simple: interoperability between devices using industry standards. To that end, the Forum selected TCP/IP as the basis for all network connectivity. Added to TCP/IP were Web standards such as HTTP, HTML, XML, and SOAP that provided the framework for device discovery, device and services description, control, and presentation.



With the core architecture defined, the Forum established a series of working groups to define device and service profiles for specific device categories. These categories include Audio/Video (AV), Internet Gateway Device (IGD), Printing, Scanning, Lighting Control, HVAC, and a number of others. The working groups – composed of member companies from relevant industries – delivered a series of XML schemas representing the baseline set of functions and services that each specific device type was required to support.

The most significant of the working groups efforts (at least in terms of digital media content) was the UPnP AV specification. In fact, it was so important that it became the basis for a new organization – DLNA. DLNA was formed in 2003 by 21 companies including Microsoft, Intel, HP, IBM, Sony, Philips, Toshiba, Pioneer, Motorola and Nokia, with the goal of accelerating the development and deployment of interoperable digital media devices for the home.

DLNA Device Model

The device model used by DLNA is derived from the UPnP Forum fundamental device model. This model consists of Devices, Services, and Control Points.

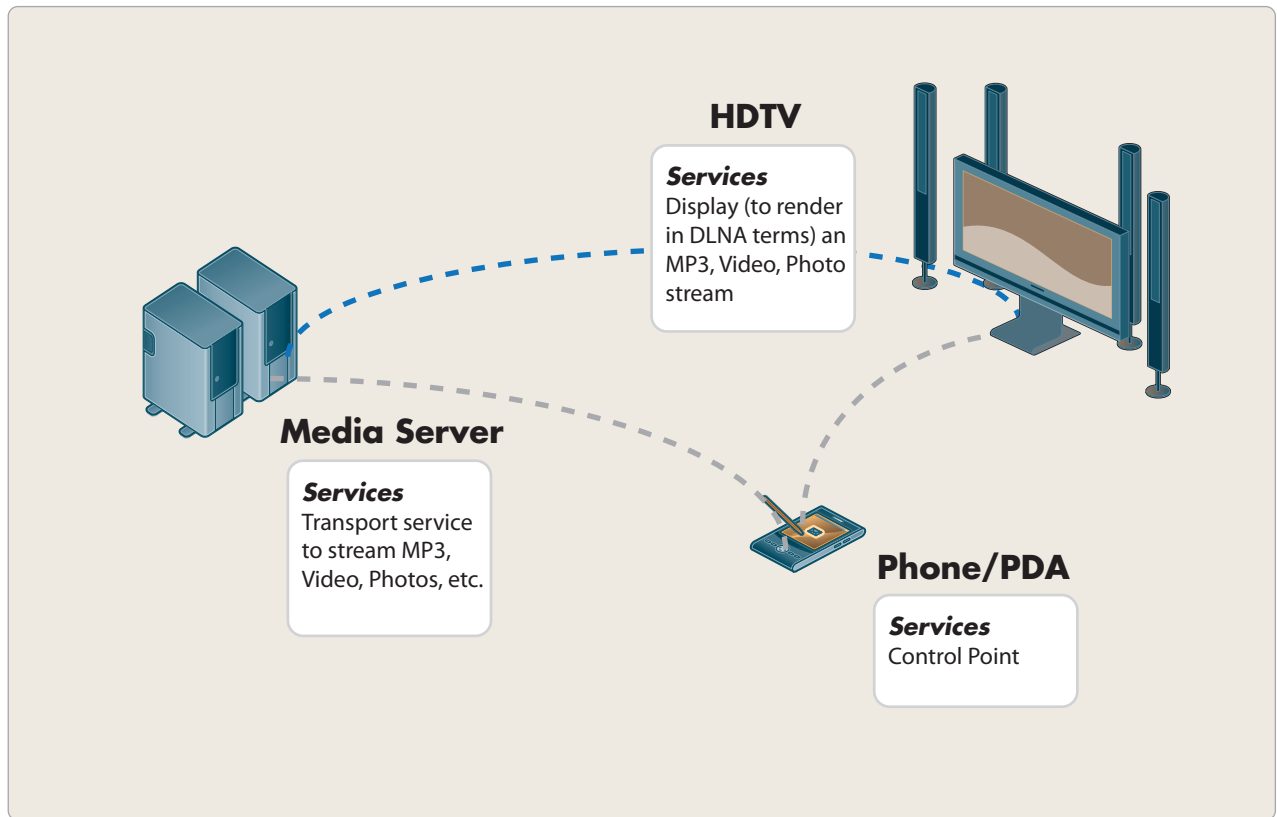
Devices are network entities that provide services and can contain other nested devices.

Services are the basic unit of control. They provide actions, and maintain status via state variables.

Control Points are network entities that are capable of discovering and controlling other devices on the network.

Figure 2.2 - DLNA devices interacting in a typical home environment.

In many cases it is very common for a DLNA *Device* to consist of multiple *Services* and potentially contain a *Control Point* as well. In Figure 2.2 above, the Media Server Device provides a transport service for streaming audio MP3 files, photos and movies stored on the internal RAID disks. The HDTV is a *Device* that provides a *Service* to display video content on the screen (in UPnP / DLNA terms it will render the content to the screen). The Phone / PDA is a *Control Point* used to tell the Media Server to stream stored content to the HDTV to display (or render) to



the screen.

With the development of the UPnP AV (and thus DLNA) specifications for digital media content devices, the basic device model was extended. All control interaction (shown as dotted gray lines) only passes between a *Control Point* and *Device(s)*, but the *Devices* themselves interact (shown as dotted blue lines) with each other to pass digital content using a non-UPnP (“out-of-band”) communications protocol.

DLNA Device Categories and Classes

To better define the characteristics of devices and the services they offer, the DLNA Interoperability Guidelines define three *Device Categories*:

- Home Network Devices (HND)
- Mobile Network Devices (MND)
- Home Infrastructure Devices (HID)

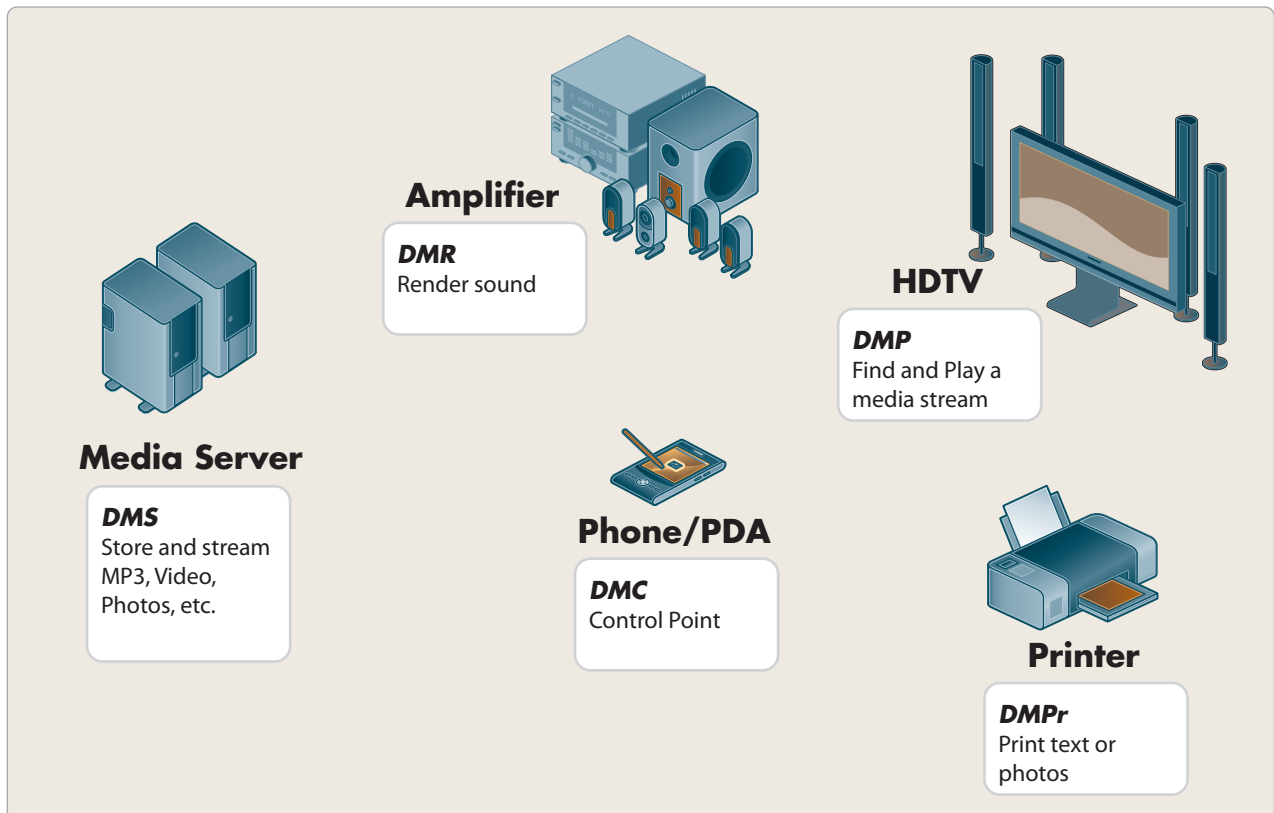
Device Categories are based on a shared set of media formats and network connections with the focus on interoperability between devices within a category. *Devices* can and often do belong to more than one *Device Category*. The underlying requirement is that the *Device* must comply with the media formats and network connectivity of both categories.

Each *Device Category* is further broken into *Device Classes*. A *Device Class* specifies the functional capabilities of a device regardless of its physical attributes. In fact, a single physical device can, and frequently does incorporate multiple *Device Classes*. For interoperability, DLNA performs device certification at the *Device Class* level. All DLNA Certified™ devices must comply with all the requirements of the *Device Class(es)* that they belong.

The Home Network Device (HND) category is made up of five *Device Classes* that are in use in the home network, and rely on the same media formats and network connectivity requirements.

- Digital Media Server (DMS)
- Digital Media Renderer (DMR)
- Digital Media Controller (DMC)
- Digital Media Printer (DMPr)
- Digital Media Player (DMP)

Figure 2.3 - DLNA Home Networking Device examples



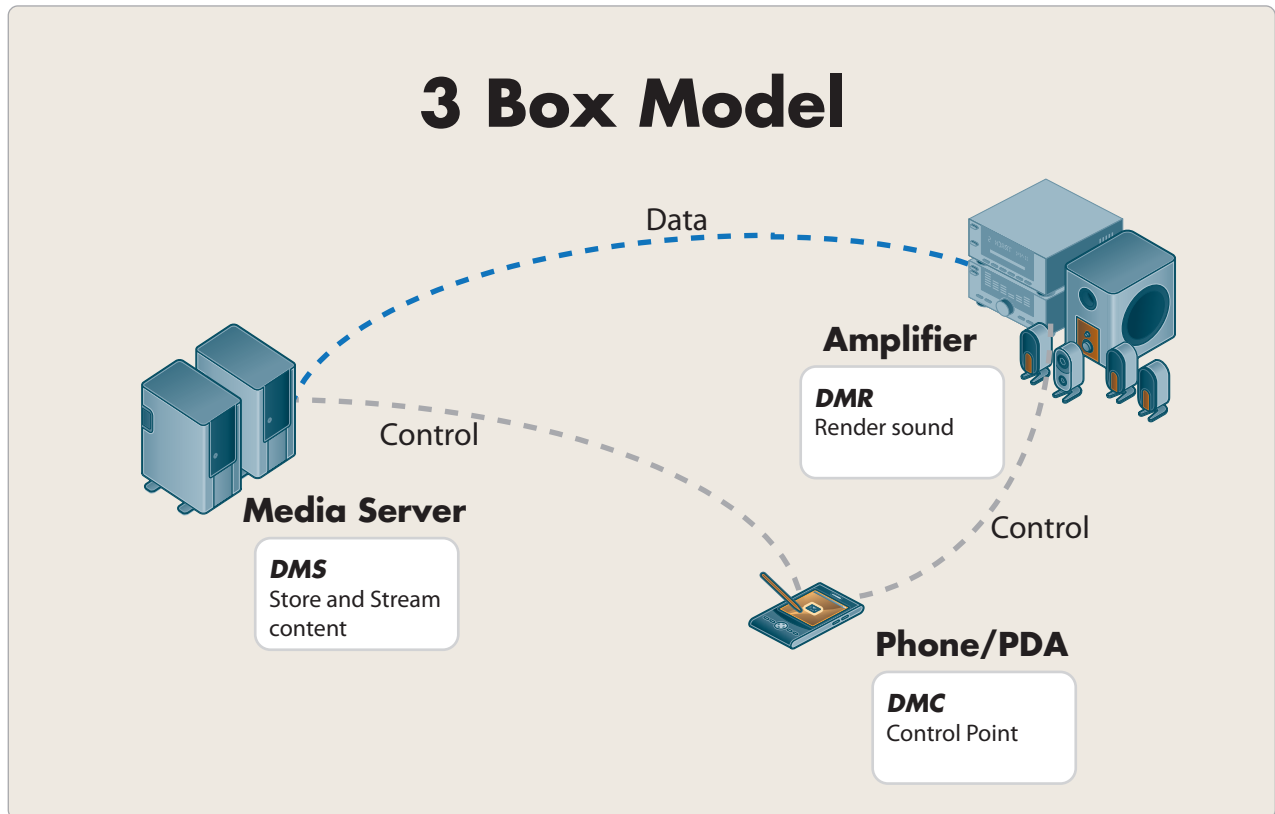


Figure 2.4 - DLNA 3 Box Model with a DMS, DMC and DMR

In the 3 Box Model (Figure 2.4) a consumer utilizes a DMC to discover DLNA devices on a home network. The DMC is used to browse and select content via a user interface (UI). The consumer continues by means of the DMC to select where the content will be played. Then, with the standards based DLNA architecture, the devices automatically connect and the content is rendered (played) for the user. The five device classes within the HND category bring this about, however there are a few characteristics worth noting.

In an effort to initially simplify DLNA architecture, two specific models or scenarios were defined and utilized by manufacturers of DLNA Certified products. The use of these scenarios bring to light various details about a DMP, DMR and DMC and how they interact in a DLNA home networking environment.

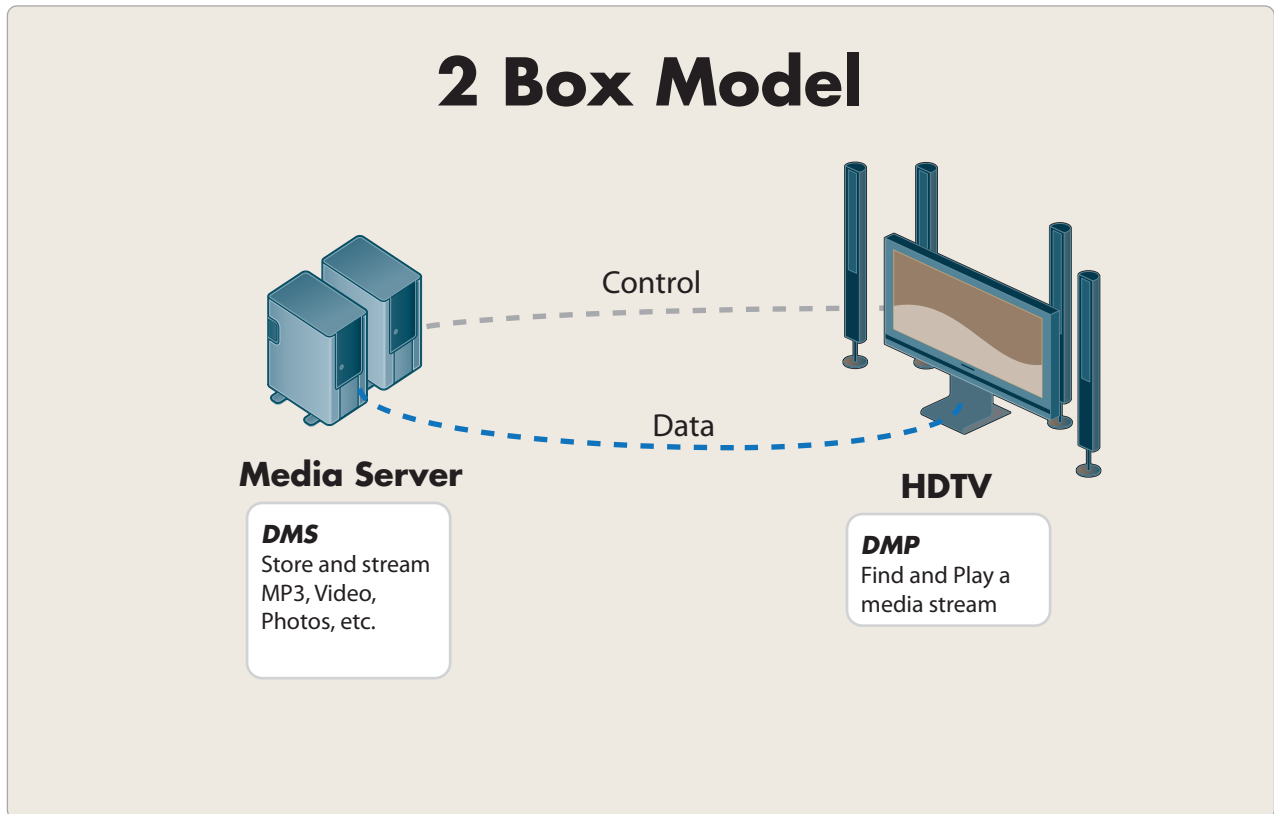


Figure 2.5 - DLNA 2 Box Model with a DMS and DMP

Figure 2.5 illustrates the DLNA 2 Box Model. The 2 Box model can be thought of as a special case of the 3 box implementation. In this use scenario, an HDTV acting as a DMP provides a consumer with an elegant user interface (UI) to find, browse, select and eventually view digital media. A DMP can be thought of as a combination of a DMC and DMR with a few additional characteristics. The DMC provides the control interface for finding, browsing and selecting content. The DMR provides the ability to render (or play) content to the wide screen TV. In the 3 Box model a DMC can discover and control (or cause) content to be rendered (or played) from any DMS to any DMR. In the case of a DMP and 2 Box Model, the DMC part of the DMP always selects the local DMR as the target renderer for all content. In addition the DMR in the 3 Box Model is always discoverable by other DLNA devices. The DMR within a DMP in the 2 Box Model is not discoverable (although some vendors also allow this DMR to be recognized outside of the DMP entity).

The Mobile Handheld Device (MHD) category is made up of five Device Classes that share the same usages models as the HND Device Category, but have different requirements for media format and network connectivity.

- Mobile Digital Media Server (M-DMS)
- Mobile Digital Media Controller (M-DMC)
- Mobile Digital Media Player (M-DMP)
- Mobile Digital Media Uploader (M-DMU)
- Mobile Digital Media Downloader (M-DMD)

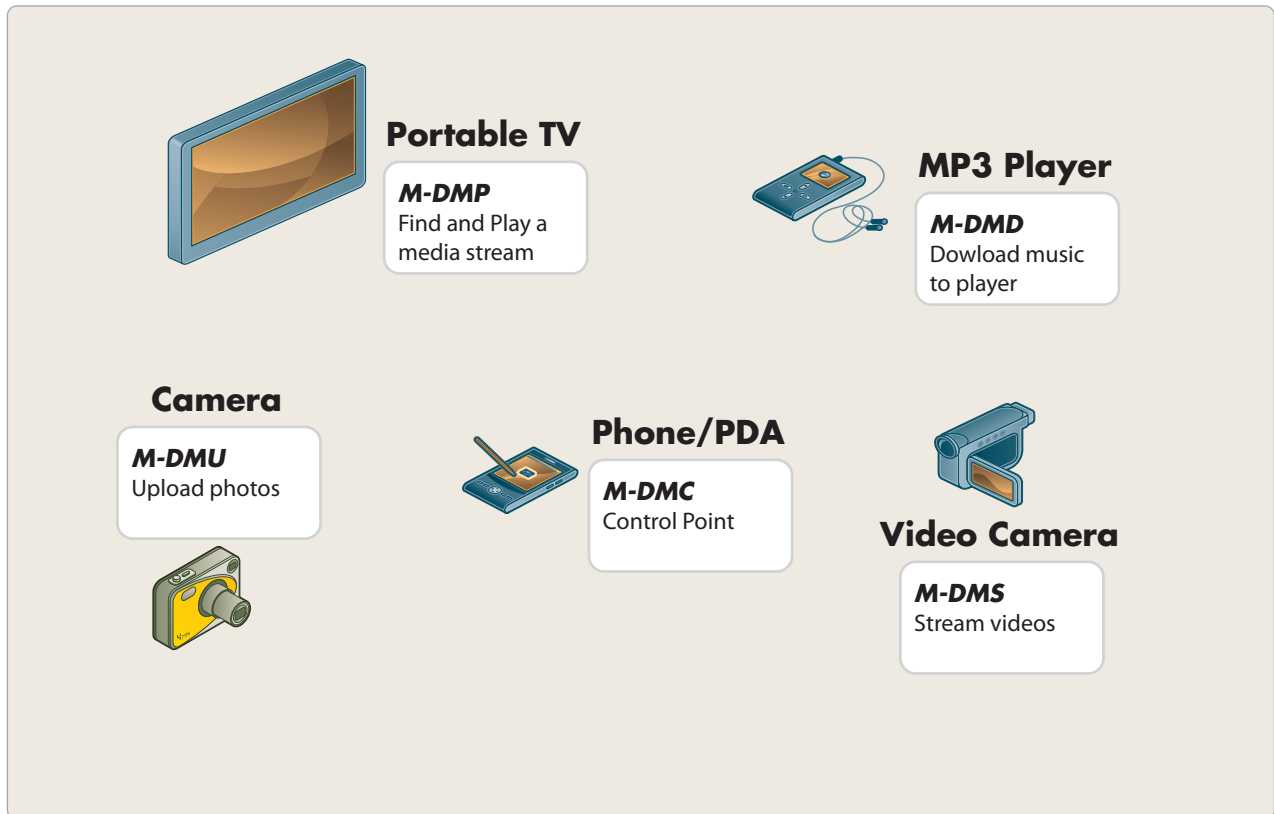


Figure 2.6 - DLNA Mobile Handheld Devices

Figure 2.6 illustrates the types of DLNA devices found within the Mobile Handheld Device category. It is important to note that just as the DMP could not be discovered in the HND the M-DMP Device Class is also undiscoverable by other DLNA devices.

The Home Infrastructure Device (HID) category is made up of two Device Classes. These devices are intended to allow HNDs and MHDs to interoperate.

- Mobile Network Connectivity Function (M-NCF)
- Media Interoperability Unit (MIU)

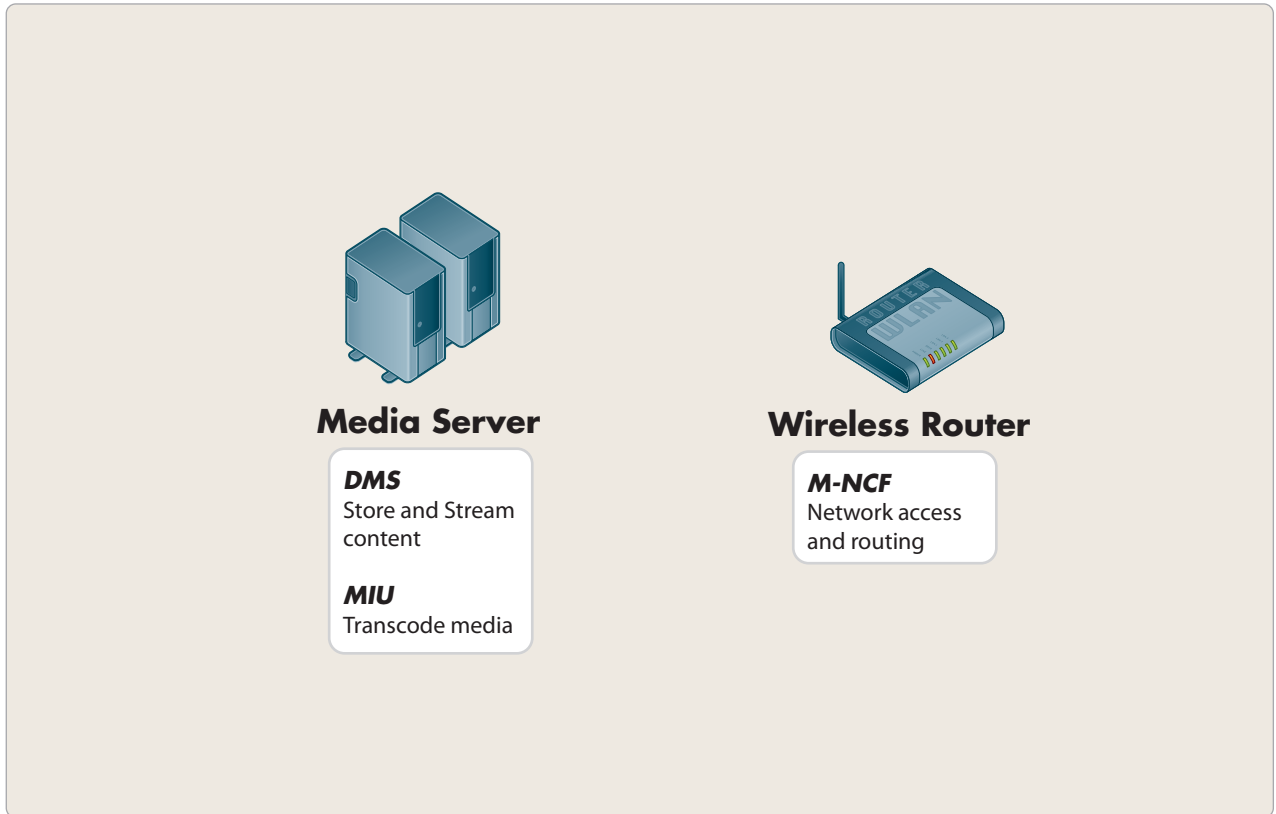


Figure 2.7- DLNA Home Infrastructure Device examples

Figure 2.7 illustrates examples of HID Devices in a home network. The Media Server is a member of the HND category as a DMS Device Class, this example shows the Media Server is also a member of the HID Device Category as a MIU Device Class. The MIU Device class provides a vital role in transcoding stored digital content into formats that mobile devices can consume. The M-NCF provides necessary access, routine and bridging functions to seamlessly connect mobile devices to a home network.

DLNA Architecture

Figure 2.8 illustrates the functional components of DLNA 1.5 as it relates to the Interoperability Guidelines network architecture. Each of the functional components are briefly discussed in the following sub-sections.

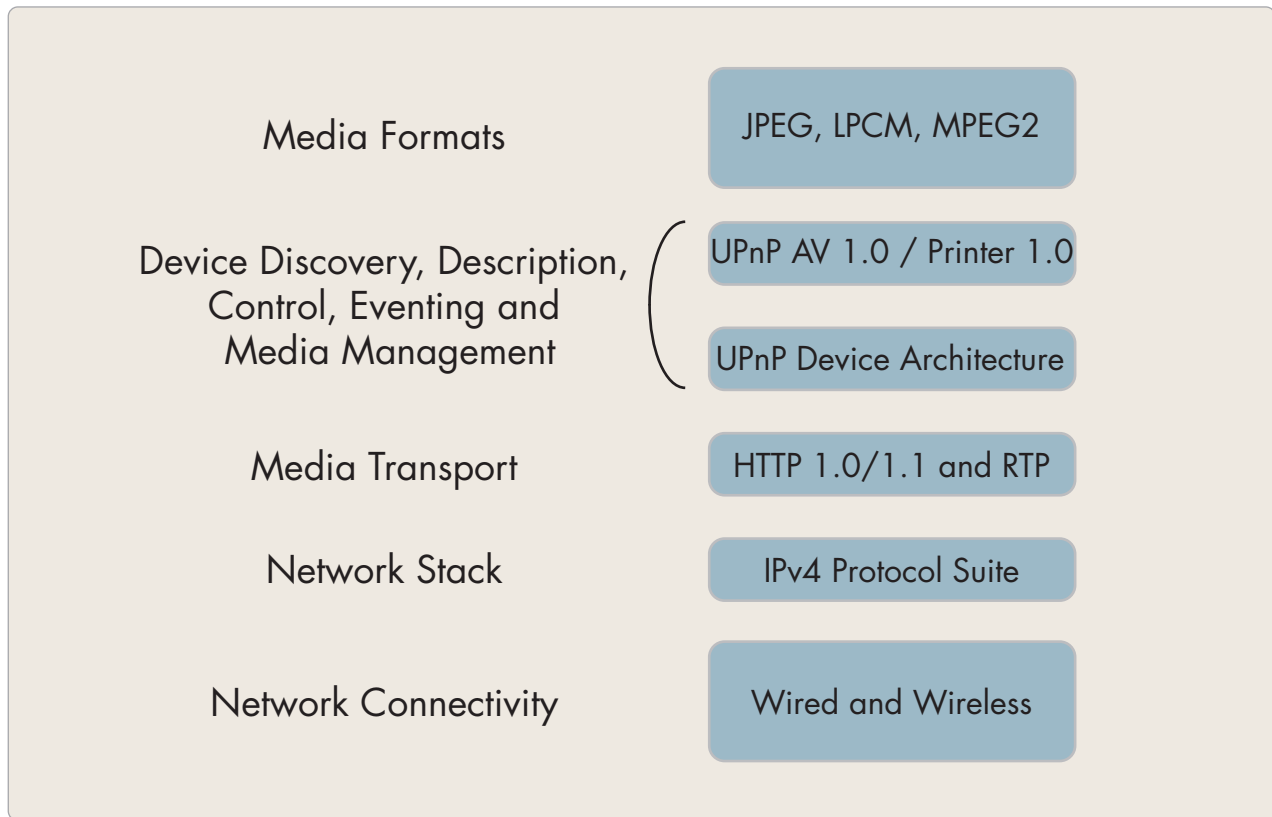


Figure 2.8 - DLNA Functional Component Architecture

DLNA Media Formats

Media Formats describe how digital content is encoded and formatted for each of the three *Media Classes*:

- Image
- Audio
- Audio Visual

Media format profiles are very explicit, with attributes, parameters, system, and compression level details defined in sufficient detail to ensure interoperability between DLNA Certified devices. The present set of *Media Formats* that must be supported by each specific device in a *Media Class* are listed in the Table 2.1. Optional media formats are also defined. The *Interoperability Guidelines* provide specific rules about using optional formats between compatible devices in addition to conversion between optional and mandatory formats.

Table 2.1 - Media Class mandatory and optional formats

Media Class	Mandatory Formats	Optional Formats
<i>Image</i>	JPEG	PNG, GIF, TIFF
<i>Audio</i>	LPCM	AC3, AAC, MP3, WMA9, ATRAC3plus
<i>Video</i>	MPEG2	MPEG1, MPEG4, VC1, MPV1

The focus on *Media Formats* is a key distinguishing factor between the UPnP Forum and DLNA. The UPnP Forum focused on achieving device interoperability, which was accomplished. But the lack of prescribed media profiles prevented the UPnP architecture from delivering media interoperability and led to the founding of DLNA.

DLNA Device Discovery, Description, Control, Eventing and Media Management

Device Discovery, Description, Control and *Eventing* enable a device on the home network to discover the presence and capabilities of other devices on the network and collaborate with these devices in a uniform and consistent manner. DLNA incorporates the UPnP Forum Device Architecture 1.0 as the basis for its device discovery and control.

Device Discovery

When a device is added to the network, the UPnP discovery protocol allows that device to advertise its services to control points on the network. Similarly, when a control point is added to the network, the UPnP discovery protocol allows the control point to search for devices of interest on the network. The message exchanged in both cases is a discovery message containing a few, essential specifics about the device, its services (such as the device type), an identifier, and an HTTP URL to access for more detailed information. The UPnP discovery protocol is based on the Simple Service Discovery Protocol (SSDP).

Description

After a control point has discovered a device, the control point needs to find out more about the device. For the control point to learn more about the device and its capabilities, or to interact with the device, the control point must retrieve the device's *Description* using the URL provided by the device in the discovery message. This URL points to a UPnP description document that is expressed in XML and includes vendor-specific information such as the model name and number, serial number, manufacturer name, URLs to vendor-specific web sites and URLs for control, eventing, and presentation.

The UPnP description document includes a list of *Device Services* provided by the device. For each *Device Service*, the *Description* includes a list of actions for the service, and arguments for each. The *Description* of a *Device Service* also includes a list of variables that reflect the state of the device. These variables are described in terms of their data type, range, and event characteristics.

Control

After a control point has retrieved the *Description* of the device, the control point can invoke actions that are supported by the device. To do this, a control point sends a suitable control message to the control URL for the service. Control messages are expressed in XML using the Simple Object Access Protocol (SOAP). Control actions are like function calls, and the device service will return action-specific values in response to the control message. The effects of the action may also be reflected by changes in the variables that describe the run-time state of the device.

Eventing

The UPnP description document for a service includes a list of variables that represent the state of the device. The device publishes updates whenever there is a change in the value of any evented variable. A control point may

subscribe to receive this information by sending a GENA (General Event Notification Architecture) message. The device publishes updates by sending event messages. Event messages contain the names of one or more state variables and their current values. These messages are expressed in XML and sent using HTTP.

A special initial event message is sent when a control point first subscribes. This initial event message contains the names and values for all evented variables and allows the subscriber to initialize its model of the state of the service. To support scenarios with multiple control points, eventing is designed to keep all control points equally informed about the state of the device. This means that all subscribers are sent all event messages, and subscribers receive event messages for all evented variables that have changed.

Presentation

The presentation capabilities of a device enable HTML-based management of the device by the end user using a standard web browser. A control point can obtain the entry presentation URL from the device description document, retrieve the entry page from the URL, load the page into a browser, and start the user management of the device.

Media Management

Media Management enables devices and applications to identify, manage, and distribute digital media content across network devices. The *Interoperability Guidelines* incorporate the UPnP Forum AV technology as the basis for DLNA *Media Management*. There are four *Device Services* provided by this technology:

- Content Directory
- Connection Manager
- AV Transport
- Rendering Control

Content Directory

The *Content Directory* service provides a mechanism for each content server on the network to provide a uniform directory of all its available content to any interested devices on the network. Every content server must have an instance of this service.

This service might provide a list of songs stored on an MP3 player, still-images comprising various slide shows stored on a PC, movies stored in a DVD jukebox, TV shows currently being broadcast by a Set-top Box, songs stored in a media server, TV programs that had been downloaded to a PVR, photos stored in a digital camera, and many more. Nearly any type of content can be listed via the *Content Directory* service, even for devices that support multiple types of content. The information about the content (metadata) returned by the *Content Directory* service includes properties such as its name, artist, creation date, size, etc. In addition, the metadata also indicates the transfer protocols and data formats that are supported for each piece of content on the server. This information is used by the *Control Point* to determine if a given Media Renderer is capable of rendering the content in its current format or if some type of transcoding is required.

Connection Manager

The Connection Manager service determines how the digital media content can be transferred between two devices on the network. Each device that sends or receives content must implement the *Connection Manager* service. This service provides a mechanism for devices to:

- To determine whether a content item can be played on a specific device
- Set up and tear down connections between devices
- Discover information about current transfers in the network

AV Transport

The *AV Transport* service enables control over the “playback” of audio and video streams including the ability to Stop, Pause, Seek, etc. This service type defines a common model for *AV Transport* control suitable for a generic user interface. It can be used to control a wide variety of disc, tape, and solid-state media devices such as DVD/Bluray/CD players, VCRs, and MP3 players. Depending on the supported transfer protocols and data formats, this service may or may not be implemented.

Although most media will be sent across the network as data it may be more efficient to transfer the media data stream using other means. An example is when a personal video recorder is the DMS and a HDTV is the DMR. An Ethernet connection would not be as efficient as an HDMI or component video connection. Using a transfer medium that is not part of the TCP/IP network is called an “out of band” transfer. These transfers are not defined by the UPnP AV specification but are recommended and supported by the manufacturer of the media equipment.

Rendering Control

Most rendering devices contain a number of dynamically configurable attributes that affect how the current content is rendered. For example, video devices, such as HDTVs, allow user control of display characteristics such as brightness and contrast, while audio devices allow control of audio characteristics such as volume, balance, and equalizer settings. The *Rendering Control* service provides control points with the ability to query and adjust any rendering attribute that the device supports.

The *Rendering Control* service enables a control point to:

- Discover the attributes supported by the device.
- Retrieve the current setting of any supported attribute.
- Change the setting of any modifiable attribute.
- Restore the settings defined by a named preset.

DLNA Media Transport

Media Transport defines how content moves across the network. DLNA devices that send or receive any media content via the network must support HTTP 1.1 (including chunked transfer encoding, persistent connections, and pipelining) as the baseline transport mechanism. In addition, Real-time Transport Protocol (RTP) is available as an optional media transport protocol.

Network Stack

The basis for the DLNA *Network Stack* is TCP/IP v4. Every device must implement a DHCP client, and search for a DHCP server when first connected to the network. If a DHCP server is discovered, the device must use the IP address assigned by the server. If no DHCP server is discovered, the device must use Auto-IP to generate a link-local IP address.

Auto-IP uses an implementation dependent algorithm to generate an address in the 169.254/16 range. The first and last 256 addresses in this range are reserved and must not be used. After developing an address, the device must determine if the address is available by using an ARP probe. If the device receives a response, the address is assumed to be in use and the device must generate and test a new IP address.

An Auto-IP configured device must periodically check for the presence of a DHCP server. If a DHCP server is discovered, the device must switch to the IP address allocated to it by the DHCP server. In order to switch between IP addresses, the device must cancel any outstanding UPnP *Discovery* advertisements and re-issue them under the new address.

In addition to IP addressing, UPnP makes extensive use of both the UDP and TCP protocols. *Discovery* is implemented via an HTTP Multicast over UDP. This method is used by devices to advertise their presence to the network and by control points to discover what devices exist on the network. Description, control, and eventing services are delivered via HTTP over TCP.

Network Connectivity

Three network connection technologies are incorporated in the DLNA 1.5 Interoperability Guidelines: 10Base-T and 100Base-T Ethernet (802.3i / 802.3u) for wired connections, WiFi (802.11a / 802.11b / 802.11g) for wireless connections, and Bluetooth for wireless connections for mobile handheld devices such as cell phones and PDAs. Additional network connections such as 1000Base-T Ethernet (802.3ab), WiFi (802.11n) and most Multimedia Over Coax Alliance (MoCA) will be added to the Guidelines in the future. It should also be noted that many other networking technologies such as LonWorks, CeBus, X-10, and Universal Powerline Bus (UPB) could be supported via UPnP Bridges.

Link Protection and Digital Rights Management (DRM)

When commercial content is made available for consumer electronics and mobile devices, it must be protected from unauthorized copying and use. Consumers now expect the capability to store, transfer and use their purchased content on any device at any location connected to wired or wireless digital networks. At this time UPnP and Digital Living Network Alliance (DLNA) are still working to provide a robust and versatile DRM solution capable of interacting with the breadth of proprietary solutions already installed. However, UPnP and DLNA along with many large industry players have approved DTCP-IP and Windows Media Digital Rights Management - Network Device (WMDRM-ND) technologies for the transport of protected content.

Certification

In order to ensure interoperability between DLNA devices, DLNA developed and manages a comprehensive

certification program. Vendor products that successfully complete certification are awarded the DLNA Certified accreditation. This lets consumers know the product is fully DLNA compliant and interoperates with other DLNA Certified products. Look for the DLNA Certified logo on products and product packaging.



The initial step in obtaining certification calls for the manufacturer of the product to subject the device to testing utilizing the DLNA's Conformance Test Tool (CTT). The CTT is a suite of tests that are run by the vendor against the product, and validate the devices' compliance with DLNA standards. The test harness for the CTT is a single Windows PC with the device under test connected via a DLNA defined network connection technology (Ethernet, WiFi, Bluetooth). When the device successfully passes the CTT as determined by the CTT's log file, it can begin the formal DLNA certification process.

The formal certification process entails submitting the CTT log and a product's UIC certificate (from UPnP) to DLNA. Next step is scheduling a test session with one of the Independent Certification Vendors (ICV) approved by DLNA. The ICV will test the submitted product per DLNA's Certification Test Plan (CTP) against 3 reference devices of the appropriate device class. For example, the ICV would test a DMP device against 3 DMS reference devices, while a DMC would be tested against 3 DMR and 3 DMS reference devices.

In addition to its formal certification program, DLNA conducts "plugfests" (interoperability workshops) on a regular basis. The plugfests are held each calendar quarter in various locations around the world in order to allow maximum participation from device vendors across the globe. These plugfests provide DLNA member companies the opportunity to test products under development against other member's products using DLNA test tools, and are an excellent dress rehearsal for DLNA certification testing.

CHAPTER 3

RDMC GETTING STARTED

RDMC 101 - Getting Started

Thank you for the purchasing the Allegro Reference Digital Media Controller 101 (RDMC 101). The RDMC 101 is a Digital Living Network Alliance (DLNA) certified reference implementation of a Digital Media Controller (DMC) and meets all DLNA 1.5 interface guidelines as a media controller. With the RDMC 101 you can query a local network to discover and control DLNA devices. Ultimately, the RDMC 101 enables you to connect DLNA Servers with DLNA Renders and control how content is transferred between the two.

The following manual provides detailed information on setup and use of the RDMC 101 in a DLNA 1.5 environment.

RDMC 101 - Getting to Know the RDMC 101

As delivered, the RDMC 101 includes the following items in the box (Figure 3.1):

- RDMC 101 - Quickstart Guide
- RDMC 101 - Unit
- Power Supply - Output 12V/2.5 Amp

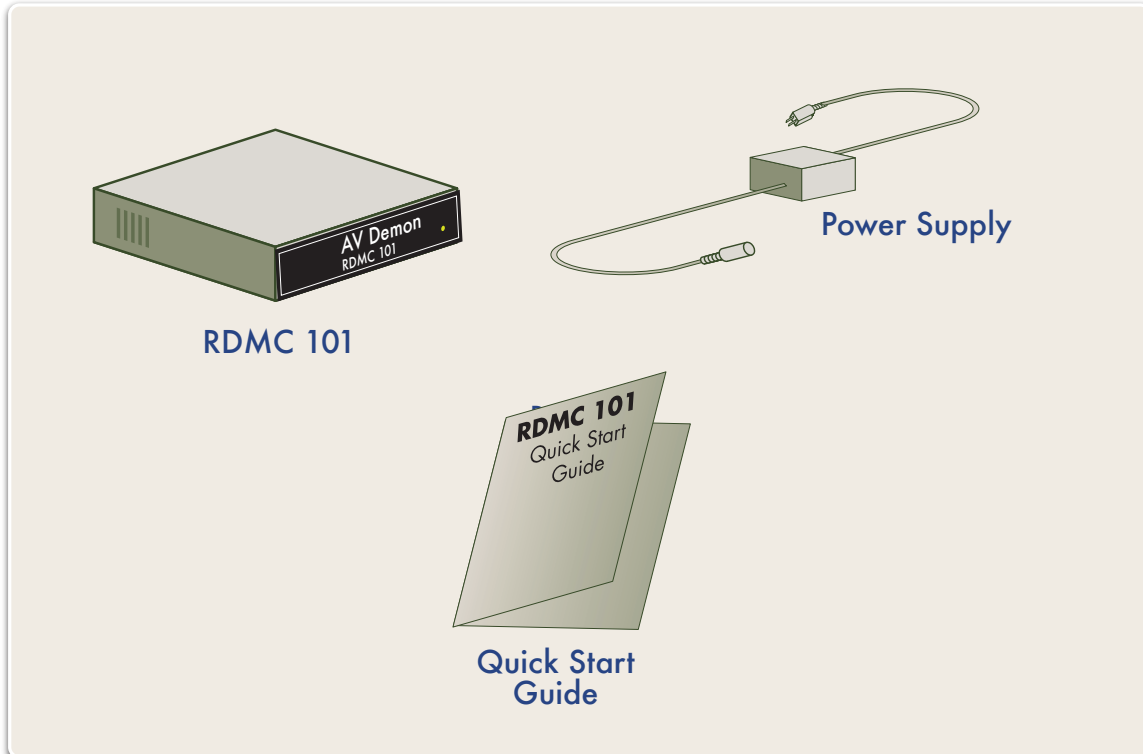


Figure 3.1 - RDMC 101 box content

Overall, the RDMC 101 is a small, yet powerful device. Measuring only 1.34" x 5" x 3.5" and weighing less than 1 pound, the RDMC 101 is an ARM based embedded system running a commercial RTOS and TCP/IP protocol stack along with the RDMC 101 application.

The RDMC 101 has a single indicator on the front of the device that is illuminated when power is applied to the device and the power switch is in the "ON" position (Figure 3.2).



Figure 3.2 - RDMC 101 Front Panel

1. Power Led - The indicator will light up green and remain on while power is connected and the power switch is in the "ON" position.

Figure 3.3 shows the back panel of the RDMC 101.

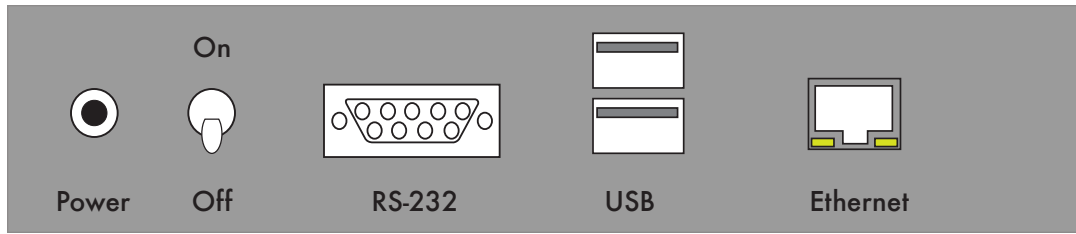


Figure 3.3 - RDMC 101 Back Panel

1. Power Connector - Only use the Allegro's provided power supply with the RDMC 101
2. Power Switch - Down Off, Up On
3. RS-232 Connector - Unused and reserved for future function - do not plug anything into this connector
4. 2 USB Connectors - Unused and reserved for future function- do not plug anything into this connector
5. RJ45 Ethernet Connector

Applicable Documents

The following documents are extremely helpful when working with the RDMC 101 and give insight into the interactions of the RDMC 101 in the DLNA 1.5 environment.

“DLNA Networked Device Interoperability Guidelines, Volume 1: Architectures and Protocols”, March 2006, Digital Living Network Alliance, <http://www.dlna.org/industry/certification/guidelines/>

RDMC 101 Requirements

For the RDMC 101 to work properly in a DLNA environment, the following minimal requirements must be met:

- A minimum 100-240 V ~ 50/60Hz AC Power at 0.8 Amps
- Ethernet connection to your local network
- A computer connected to the same local network as the RDMC 101
- The computer must have either a browser (Firefox, Netscape, IE, etc.) or a Telnet client to communicate with the RDMC 101

Connecting the RDMC 101 to a Network

Figure 3.4 illustrates how to connect the RDMC 101 to your local network.

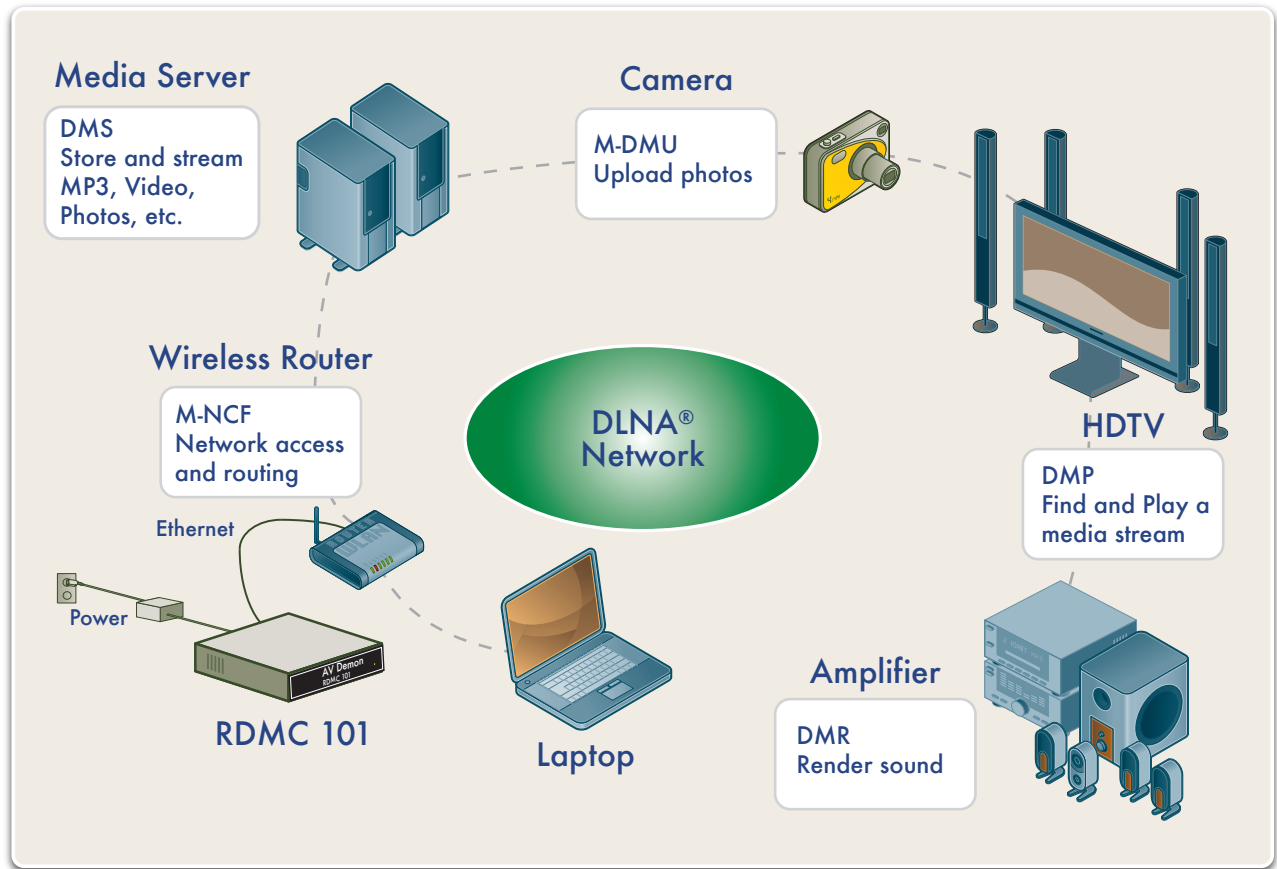


Figure 3.4 - Connecting the RDMC 101 to a Local Network

To connect the RDMC 101 to your local network, use the steps outlined below:

1. Connect one end of an Ethernet cable to the RJ45 connector on the back of the RDMC 101. Connect the other end of the Ethernet cable to a router, switch or hub in your network.
2. Connect the Allegro provided power supply to the power connector on the back of the RDMC 101. Plug the power supply into a power source.
3. Flip the power switch to the "ON" position (UP)

The Power LED on the front panel will illuminate a steady green and RDMC 101 will proceed to boot (less than 10 seconds).

At this point the RDMC 101 is fully powered and ready to control DLNA devices present on your local network. Refer to the following chapter on how to use the RDMC 101 interfaces to control DLNA devices on your local network.

CHAPTER 4

CONTROLLING DLNA DEVICES

Interacting with the RDMC 101

The RDMC 101 has two communication interfaces (Figure 4.1) for interacting with the unit: HTTP and Telnet. The Telnet interface provides a wealth of options for discovering, browsing, interrogating and controlling DLNA devices. Using a Telnet client, a user has access to very granular information and commands related to DLNA. The HTTP or web interface demonstrates the capability of using various DLNA attributes to create higher level functionality typically found in a media controller.

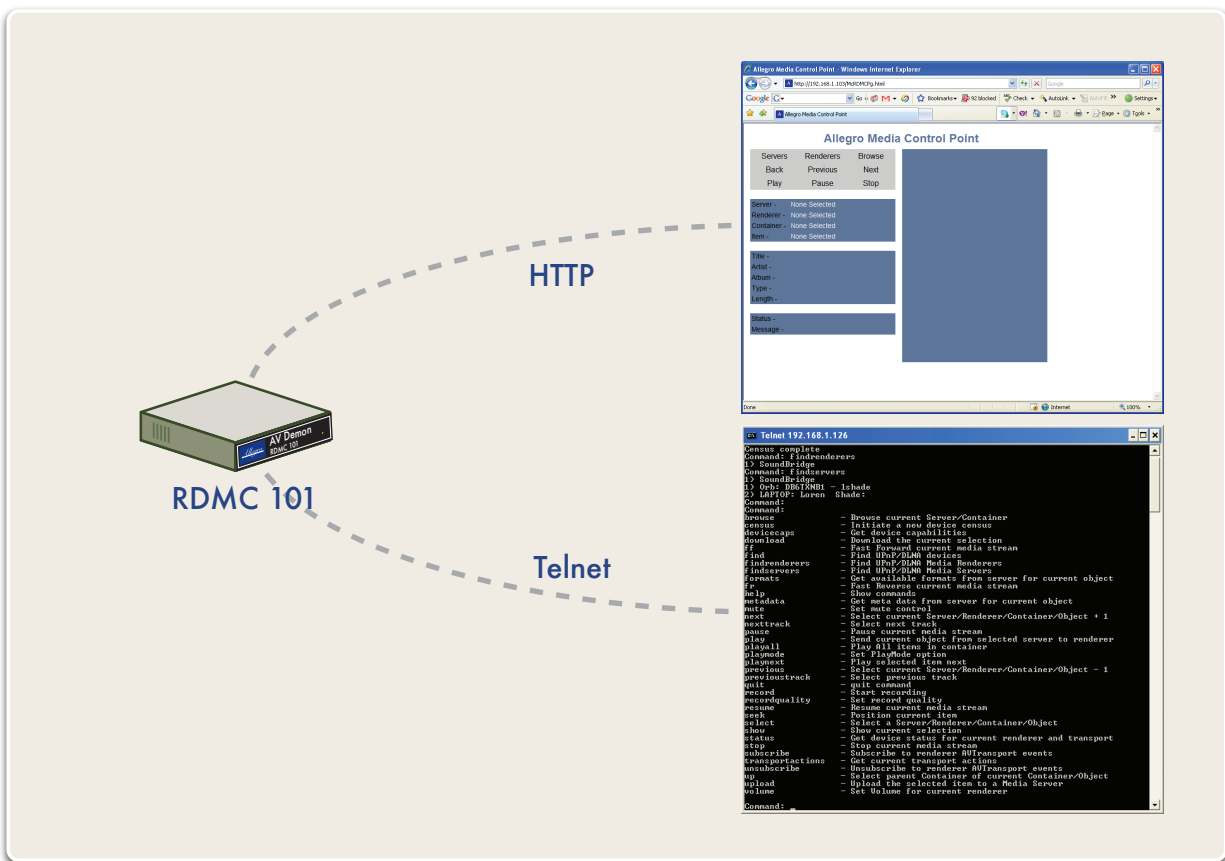


Figure 4.1 - Telnet and HTTP Interfaces for RDMC 101

RDMC 101 HTTP Interface

After connecting the RDMC 101 to your local network and turning on the power, the RDMC 101 will power up and announce itself on the network. If you are using a Windows based PC with UPnP turned on, the RDMC 101 will appear when browsing your *Network* (Figure 4.2).

RDMC101 as seen from *Network* browser

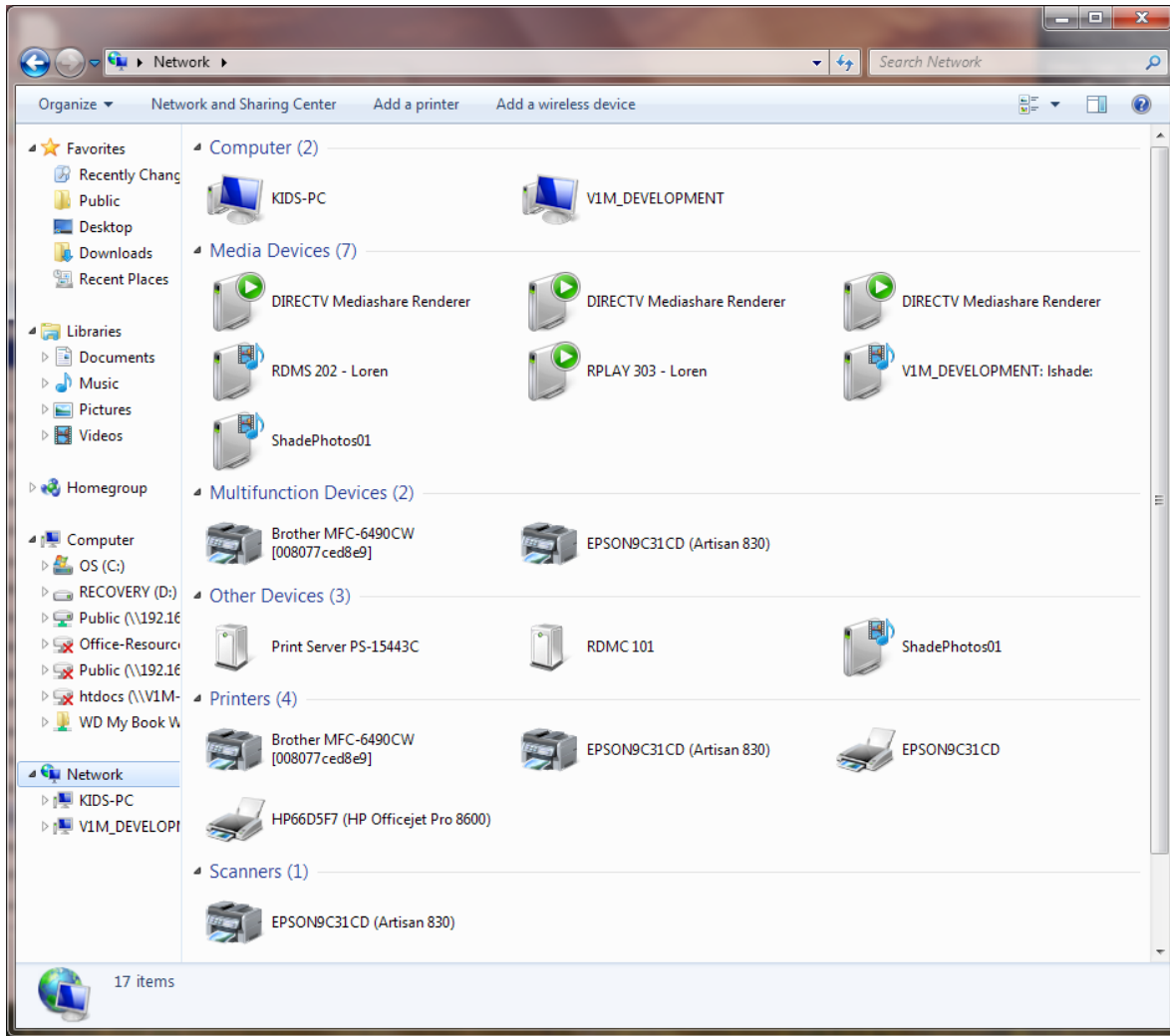


Figure 4.2 - RDMC 101 shown in My Network Places

By double clicking on the RDMC 101 icon, the RDMC 101 presentation page will appear in your web browser (Figure 4.3).

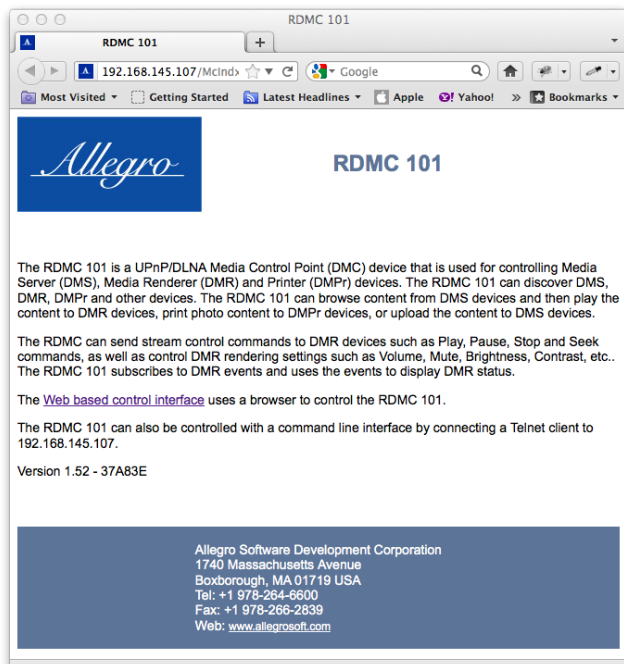


Figure 4.3 - RDMC 101 Presentation Page

After following the link to the *Web based control interface*, your browser window will display RDMC 101 Home Control Page - MDiscoverPg.html (Figure 4.4). When first displayed, this page automatically performs a search for UPnP/DLNA based devices. At anytime another device search can be commanded by clicking on **Device Search** in the upper left of the display.

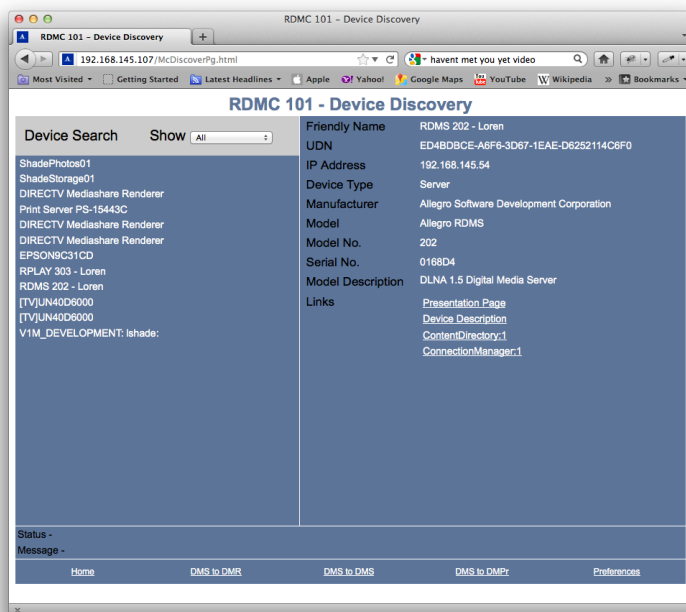


Figure 4.4 - RDMC 101 Home Control Page (McRDMCPg.html)

Search results can also be filtered using the drop down selector next to *Show*.

From the RDMC Discover page, The RDMC 101 can be used for the following types of UPnP/DLNA transactions:

- DMS to DMR – Content from a server can be directed to a renderer
- DMS to DMS – Content from a server can be uploaded to another server
- DMS to DMPr – Content from a server can be directed to a printer

RDMC 101 Preferences can also be accessed from this page.

To best describe the functional and operational characteristics, each capability is described in detail in the following sections.

DMS to DMR

After selecting the **DMS to DMR** link on the RDMC Discover page, the McControlPg.html page is shown in your web browser (Figure 4.5)

Figure 4.5 DMS to DMR Control Page

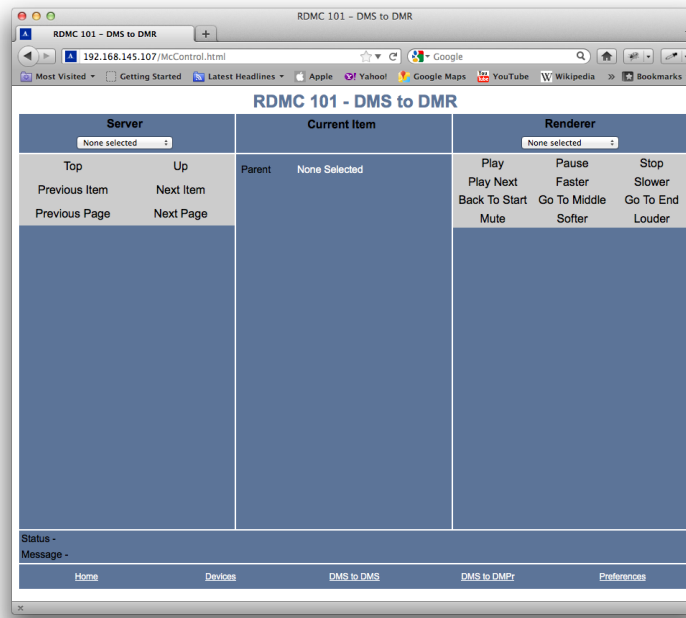


Figure 4.5 - DMS to DMR Control Page

Selecting a Server

To select a server click on the drop down menu under the Server designation in the top left portion of the page. The drop down contains all UPnP/DLNA devices that have advertised their capabilities as a server.

Once a server is selected, your web browser will display a page similar to Figure 4.6. The page shows the top level of the Content available on the selected server. Each server is unique and will likely display content different than shown in Figure 4.6. Once the top level content is presented, click on items listed to traverse the content tree to review content.

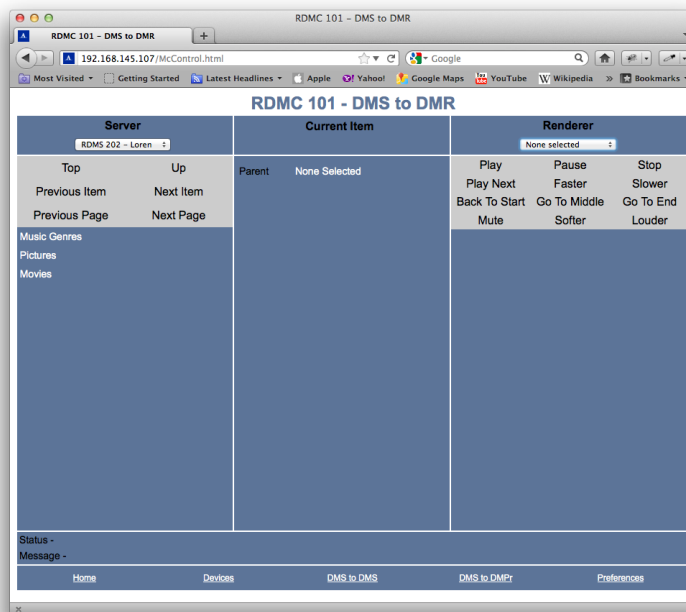


Figure 4.6 - DMS to DMR Control Page with Server Selected

Selecting a Renderer

To select a renderer click on the drop down menu under the *Renderer* designation in the top right portion of the page. The drop down contains all the UPnP/DLNA devices that have announced themselves as a renderer during the device search.

Controlling a Renderer

Under the *Renderer* section on the right side of the page, there are numerous controls to effect how a renderer operates (Figure 4.7). Each of the controls are listed below:

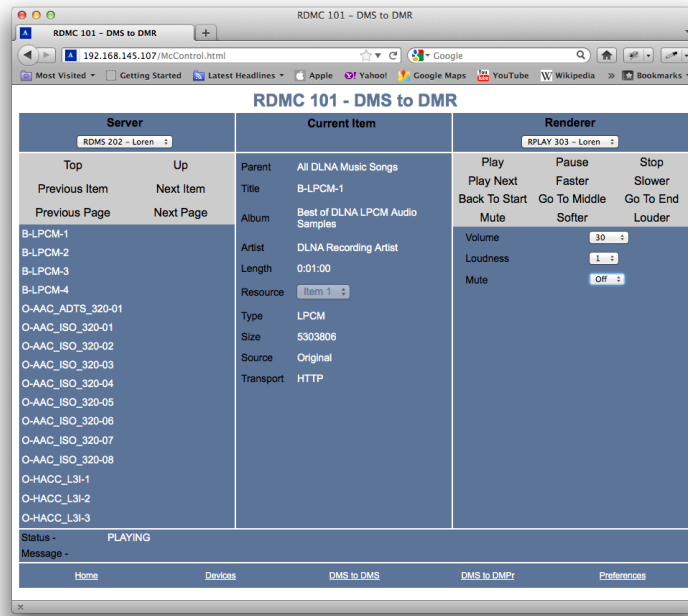


Figure 4.7 - DMS to DMR Control Pager showing a transfer in process

<i>Control</i>	<i>Definition</i>
Play	Play the selected item from the server
Pause	Pause rendering the current item
Stop	Stop rendering the current item
Play Next	Play the next item in the container
Faster	Increase the rate of playback for the current item
Slower	Decrease the rate of playback for the current item
Back to Start	Return to the beginning of the current item and render
Go To Middle	Index to the middle of the current item and render
Go To End	Index to the end of the current item and render
Mute	Mute the output
Softer	Decrease the volume
Louder	Increase the volume

Additionally there are three (3) drop down controls for Volume, Loudness and Mute.

DMS to DMS

After selecting the DMS to DMS link on the RDMC Discover page, the McUpdPg.html page is shown in your web browser (Figure 4.8). Under the Server section on the upper left section of the page use the drop down menu to select a source server to copy content from. Note, the RDMC 101 automatically filters the contents of the drop down menu to only include servers as reported from the last device search.

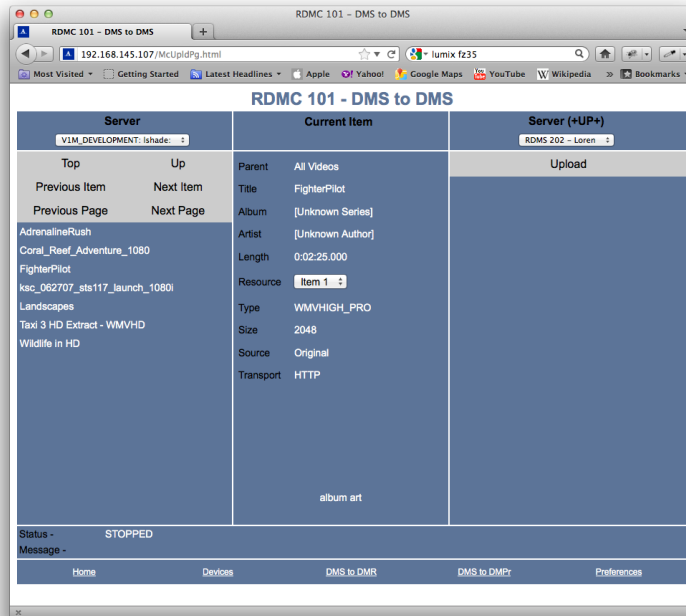


Figure 4.8 - DMS to DMS Upload Page

After a server has been selected, browse content on the source server to select an item to upload. Next select the destination server using the drop down menu under the **Server (+UP+)** section on the upper right of the page. Again note that the RDMC 101 automatically filters the list of servers to only reflect +UP+ capable servers.

Once the source server and item are selected and the destination +UP+ server is also selected, click on the **Upload** button in the upper right section of the page to start the transfer

DMS to DMP_r

After selecting the **DMS to DMP_r** link on the RDMC Discover page, the *McPrint.html* page is shown in your web browser (Figure 4.9). Under the **Server** section on the upper left section of the page use the drop down menu to select a source server with content to send to the printer.

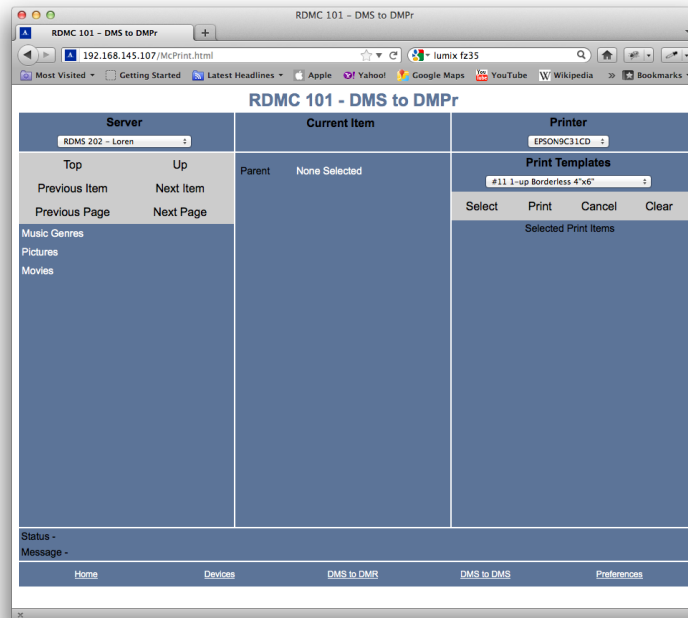


Figure 4.9 - DMS to DMPr Printer Control Page

Next, in the upper right portion of the page, select the printer to print to. Note, the RDMC 101 automatically filters the drop down list to only include devices that have announce themselves as printers during the last device search.

Once a printer is selected, the drop down menu below *Print Templates* will be populated with all printing templates supported by the printer. Select a print template from the drop down menu to communicate print configuration to the printer. Next, browse and select content from the source server to be printed. Be sure to click on *Select* to populate the content into the previously selected print template. This is especially important if the print template support multiple pieces of content. Once all content is populated into the print template, click on *Print* to send the contents and selection of the print template to the printer (Figure 4.10)

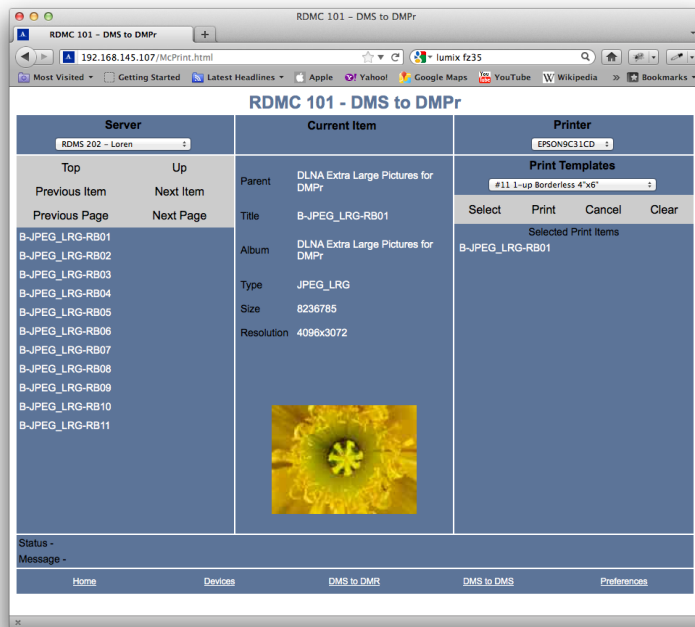


Figure 4.10 - DMS to DMPr Printer Control Page with Print Template and Content Selected

Preferences

After selecting the *Preferences* link on the RDMC Discover page, the *McPrefPg.html* page is shown in your web browser (Figure 4.11). The RDMC 101 *Preferences* page allows users to configure four (4) specific as well as check for software update and reset all settings to factory defaults.

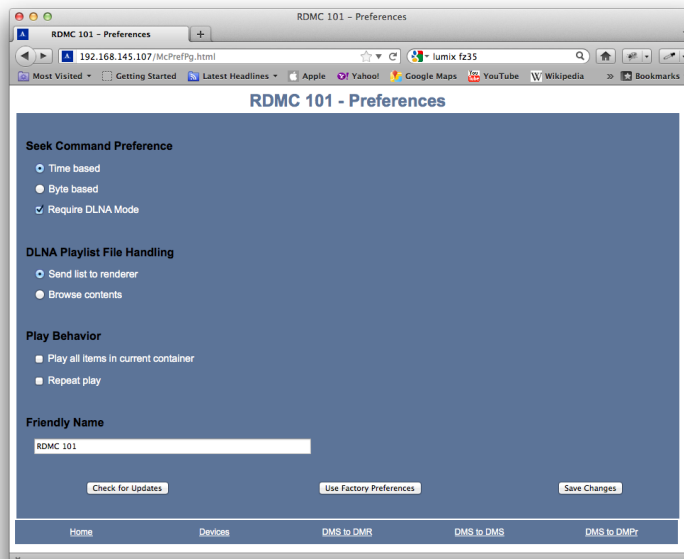


Figure 4.11 - Preferences Control Page

Seek Command Preference

This option allows users to select how seek indexes are performed. Two (2) options are presented, *Time based* and *Byte based*. *Time based* causes seek operations to reference timing metadata within the file. While this can be the most accurate it requires the device to interpret the contents of the file to acquire the needed timing information. This in turn slows seek operations. *Byte based* seek operations reference the total size of the file and perform a simple linear interpolation to seek or index to a position in the file. While this is the fastest method, due to various video compression schemes and the actual content – this method can yield inaccurate results. The user can also specify the use of DLNA mode seek. Regardless of *Time based* or *Byte based* indexing, requiring DLNA Mode places additional indexing protocol requirements on the device when performing Seek Command operations.

DLNA Playlist File Handling

DLNA offers multiple methods for handling playlists. The preference allows the user to configure if a playlist is sent to the renderer or if the RDMC creates the playlist by browsing contents.

Play Behavior

The RDMC 101 can be configured to monitor UPnP/DLNA events and automatically play all items in a container. If the ***Play all items in current container*** radio button is not selected, the RDMC 101 only renders the selected item and then stops. Users can also configure the RDMC 101 to ***Repeat play*** an item or all items in the container.

Friendly Name

Setting the Friendly Name allows the user to set a unique name for the RDMC 101.

Check for Updates

After clicking on the ***Check for Updates*** button in the lower left portion of the Preferences page your browser will display the *GetUpdate.html* page (Figure 4.12). This page displays the current firmware version and any versions available for the RDMC 101 via the Allegro update center.

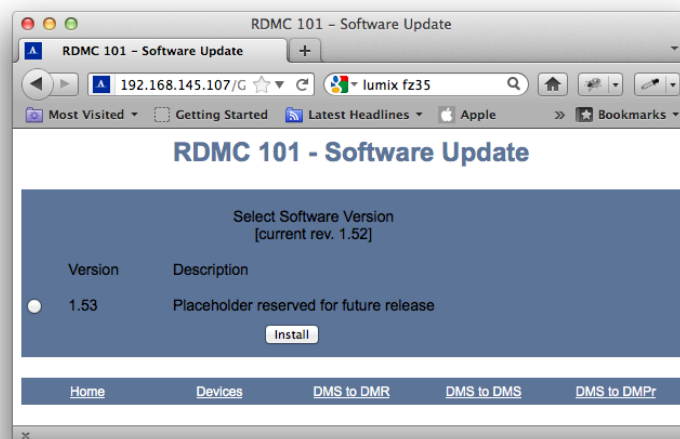


Figure 4.12 - Check for Updates

Use Factory Preferences

At any point, users can elect to reset all preferences to Factory Defaults by clicking on the Use *Factory Preferences* button.

RDMC 101 Telnet Interface

The RDMC 101 support as a rich telnet interface. The commands available via this interface offer a finer grained control and interaction with DLNA devices on the network. A summary of all the Telnet commands is provided below (Table 4.1). Each command is discussed in detail in the following text.

Table 4.1 - Summary of Telnet commands for RDMC 101

Command	Description
browse	Browse current Server/Container
census	Initiate a new device census
devicecaps	Get device capabilities
download	Download current selection
ff	Fast Forward current media stream
find	Find UPnP/DLNA devices
findrenderers	Find UPnP/DLNA Media Renderers
findservers	Find UPnP/DLNA Media Servers
formats	Get available formats from server for current object
fr	Fast Reverse current media stream
help	Show commands
metadata	Get meta data from server for current object
mute	Set mute control
name	Change the friendly name of the RDMC unit
next	Select current Server/Renderer/Container/Object + 1
nexttrack	Select next track
pause	Pause current object from selected server to renderer
play	Send current object from selected server to renderer
playall	Play ALL items in container
playmode	Set Playmode option
playnext	Play selected item next
previous	Select current Server/Renderer/Container/Object – 1
previoustrack	Select previous track
quit	quit command
record	Start recording
recordquality	Set record quality
resume	Resume current media stream
seek	Position current item
select	Select a Server/Renderer/Container/Object
show	Show current selection
status	Get device status for current renderer and transport
stop	Stop current media stream
subscribe	Subscribe to renderer AVTransport events
transportactions	Get current transport actions
unsubscribe	Unsubscribe to renderer AVTransport events
up	Select parent Container of current Container/Object
upload	Upload the selected item to a Media Server
volume	Set Volume for current renderer

Command

browse

Description

The `browse` command will display the contents of the currently selected media server. If you specify an index number as an argument for the command, the contents will be display starting with the item specified by the index. For example, if a directory contains 15 items, you can display items 10 through 15 by using the command `browse 10`.

Syntax

`browse <n>`

Inputs:

n - Index number

Outputs:

Contents of current Server container

Example:

Command: `find`

```
1) SoundBridge           ;Renderer
2) LAPTOP: Loren Shade:  ;Server
3) Orb: DB6TXNB1 - lshade ;Server
```

```
Command: select 2
```

```
Current Server - LAPTOP: Loren Shade:
```

Command: `browse`

```
1) Music                 ;Container (10)
2) Video                 ;Container (7)
3) Pictures              ;Container (7)
4) Playlists            ;Container (2)
```

Command

census

Description

Performs a census to determine what DLNA devices are available. The RDMC 101 will not respond to commands while a `census` is in process (approx 5 seconds). The RDMC will search the network for all media servers and media renderers. To display the list of media servers and media renderers found by this command, use the `find`, `findservers` or `findrenderers` command.

Syntax

census

Inputs:

None

Outputs:

When the census is finished the RDMC 101 will respond with `complete`. The results of the census are displayed using the `find`, `findrenderers` and `findservers` commands.

Example:

Command: census

Census complete

Command: find

```
1) Orb: DB6TXNB1 - lshade ;Server
2) LAPTOP: Loren Shade: ;Server
3) SoundBridge ;Renderer
Command:
```


Command

devicecaps

Description

Displays what capabilities the selected media renderer can process. The RDMC will send a GetDeviceCapabilities SOAP action to the media renderer.

Play Media, Record Media, and Record Quality will be displayed.

Syntax

devicecaps

Inputs:

None

Outputs:

A table show what types of media formats are supported for :

- Play Media
- Record Media
- Record Quality

Example:

Command: find

```

1) Orb: DB6TXNB1 - lshade      ;Server
2) LAPTOP: Loren Shade:      ;Server
3) SoundBridge                ;Renderer
   Command:                   select 3
   Current Renderer -         SoundBridge
   Command:                   devicecaps
   Device Capabilities command
   Play Media:                 UNKNOWN,NETWORK
   Record Media:               NOT_IMPLEMENTED
   Rec Quality:                NOT_IMPLEMENTED

```

Command:

Command

download

Description

Downloads the selected object from a media server to the RDMC 101. The RDMC 101 does not store the content, but exercises download functionality.

Syntax

download

Inputs:

None

Outputs:

After issuing the download command, the RDMC 101 responds with the URL related to the selected object as defined by the server. The RDMC 101 then commences downloading the selected content from the server. The content is not saved to the RDMC 101.

Example:

```
Command: find
1) LAPTOP: Loren Shade: ;Server
2) SoundBridge ;Renderer
3) Orb: DB6TXNB1 - lshade ;Server
Command: select 1
Current Server - LAPTOP: Loren Shade:
Command: browse
1) Music ;Container (10)
2) Video ;Container (7)
3) Pictures ;Container (7)
4) Playlists ;Container (2)
Command: select 2
Current Container - Video
Command: browse
1) All Video ;Container (3)
2) Video Genres ;Container (1)
3) Video Actors ;Container (1)
4) Series ;Container (1)
5) Rating ;Container (5)
6) Video Playlists ;Container (0)
7) Folders ;Container (1)
Command: select 1
Current Container - All Video
Command: browse
1) House 2008 113 ;video/avi
2) House 2008 114 ;video/avi
3) House 2008 115 ;video/avi
Command: select 1
Current Item - House 2008 113
Command: download
```

```
GET http://192.168.1.136:10243/WMPNSSv3/3152157190/1_ezNFRTkzMDEyLThGMzMtNDMOMiO5MzM1LT1CMjZFODBEMjiI5Nn0uMC44.avi
```

```
Command:
```

Command

ff

Description

Fast Forward the current media stream. Related commands are fr, pause, play, playall and stop.

Syntax

ff <n>

Inputs:

n -

Outputs:

None.

Example:

Command

find

Description

Displays all UPnP/DLNA devices found by the RDMC 101 during a census.

Syntax

find

Inputs:

None.

Outputs:

An enumerated list of UPnP/DLNA devices found on the local network.

Example:

```
Command:                census
Census complete
Command:                find
1) Allegro Media Server ;Server
2) LAPTOP: Loren Shade: ;Server
3) SoundBridge          ;Renderer
4) Orb: DB6TXNB1 - lshade ;Server
Command:
```

Command

findrenderers

Description

Displays all UPnP/DLNA Media Renderer devices found by the RDMC 101 during a census.

Syntax

findrenderers

Inputs:

None

Outputs:

An enumerated list of all UPnP/DLNA Media Renderer devices found on the local network.

Example:

```
Command:                census
Census complete
Command:                findrenderers
1) SoundBridge
Command:
```

Command

findservers

Description

Displays all UPnP/DLNA Media Server devices found by the RDMC 101 during a census.

Syntax

findservers

Inputs:

None.

Outputs:

An enumerated list of all UPnP/DLNA Media Server devices found on the local network.

Example:

```
Command:                census
Census complete
Command:                findservers
1) Orb:                 DB6TXNB1 - lshade
2) LAPTOP:              Loren Shade:
3) Allegro Media Server
Command:
```

Command

formats

Description

A user must first select a server and browse and select a specific object on the server. Once an object has been selected, the `formats` command displays the available formats for the selected object on the server.

Syntax

formats

Inputs:

None

Outputs:

An enumerated list of all formats for the selected object on the server . A server may store multiple copies of an object in varying formats or offer realtime transcoding

Example:

```
Command:                census
Census complete
Command:                find
1) Allegro Media Server ;Server
2) LAPTOP: Loren Shade: ;Server
3) SoundBridge          ;Renderer
4) Orb: DB6TXNB1 - lshade ;Server
Command:                select 3
Current Renderer - SoundBridge
Command:                select 2
Current Server - LAPTOP: Loren Shade:
Command:                browse
1) Music                ;Container (10)
2) Video                ;Container (7)
3) Pictures             ;Container (7)
4) Playlists           ;Container (2)
Command:                select 1
Current Container - Music
Command:                browse
1) All Music           ;Container (1417)
2) Genre               ;Container (23)
3) All Artists         ;Container (115)
4) Contributing Artists ;Container (101)
5) Album Artists       ;Container (107)
6) Composers           ;Container (492)
7) Albums              ;Container (157)
8) Rating              ;Container (5)
9) Music Playlists     ;Container (0)
10) Folders            ;Container (2)
Command:                select 1
Current Container - All Music
Command:                browse
1) "Close Call" with Asteroid ;audio/mpeg
2) "Highway Blues"          ;audio/x-ms-wma
3) 01 Track 1                ;audio/x-ms-wma
4) 1-2-3                      ;audio/mpeg
5) 5 Reasons Businesses
   Must Embrace Podcasting - Marketing Online;audio/mpeg
6) The 12 Days Of
   Christmas [Live]          ;audio/mpeg
7) 17th Century Canon       ;audio/mpeg
8) 19 Somethin'             ;audio/mpeg
9) 25 or 6 to 4             ;audio/mpeg
10) 52nd Street Theme       ;audio/mpeg
11) 100 Greatest Discoveries
    - Earth Sciences         ;audio/mpeg
12) 100 Years                ;audio/mpeg
13) About You (With All
    Of My Heart Album Version) ;audio/mpeg
```

```
14) Abside the Winter           ;audio/mpeg
15) Accidentally In Love       ;audio/mpeg
16) Across the Lines           ;audio/mpeg
17) Across the Midnight Sky    ;audio/mpeg
18) Across the Stars
    (Love Theme from Attack
     of the Clones)           ;audio/mpeg
19) An Actor's Life            ;audio/mpeg
20) Adventures [Unaltered]     ;audio/mpeg
21) After Midnight             ;audio/mpeg
22) After Ventus               ;audio/mpeg
23) After We Make Love         ;audio/mpeg
24) After You've Gone          ;audio/mpeg
Enter <browse> for more items
Command:                       select 9
Current Item - 25 or 6 to 4
Command:                       formats
Formats available for 25 or 6 to 4:
1) MP3
2) LPCM
Command:
```

Command

fr

Description

Fast reverse current media stream. Related commands are ff, pause, play, playall and stop.

Syntax

fr <n>

Inputs:

n -

Outputs:

None.

Example:

Command

help

Description

Displays all RDMC 101 commands. Help can also be obtain for each command by typing the command followed by a "?". The list of commands can also be displayed by typing a "?" at the Command: prompt.

Syntax

help

Inputs:

None.

Outputs:

A list of all supported Telnet commands.

Example:

Command: help

```
***** Device Commands *****
browse <n> <cnt> - Browse current Server/Container
find           - Find UPnP/DLNA devices
findservers   - Find UPnP/DLNA Media Servers
findrenderers - Find UPnP/DLNA Media Renderers
next          - Select current Server/Renderer/Container/Object + 1
previous      - Select current Server/Renderer/Container/Object - 1
select <n>    - Select a Server/Renderer/Container/Object
upload        - Upload the selected media item to a Media Server
download      - Download the selected media item
name <name>   - Set the device's friendly name
***** Player Commands *****
show          - Show current selection
ff            - Fast Forward current media stream
fr            - Fast Reverse current media stream
metadata      - Display current item metadata
mute <n>     - Set mute
nexttrack    - Skip to next track
pause        - Pause current media stream
play <res>   - Send current object from server to renderer
playmode <n> - Set play mode option
playall      - Play all items in the current container
playnext <res> - Play selected item next on renderer
previoustrack - Skip to previous track
record       - Start recording
resume       - Resume current media stream
seek <n>    - seek to position
stop        - Stop current media stream
up          - Select parent Container of current Container/Object
volume <n>  - Set Volume for current renderer
***** Control Commands *****
subscribe    - Subscribe to current renderer AVTransport events
unsubscribe  - Unsubscribe to current event subscription
status      - Get device status for current renderer and transport
censu       - Initiate a new device census
devicecaps  - Show device capabilities
transportactions - Show current transport actions
help        - Show these commands
Command:
```

Command

metadata

Description

Displays the metadata from the server for the selected object.

Syntax

metadata

Inputs:

None.

Outputs:

A list of the associated metadata with the current item.

Example:

```
Command: census
Census complete
Command: find
1) Orb: DB6TXNB1 - lshade ;Server
2) LAPTOP: Loren Shade: ;Server
3) SoundBridge ;Renderer
Command: select 1
Current Server - Orb: DB6TXNB1 - lshade
Command: browse
1) Music ;Container (3)
2) Pictures ;Container (2)
3) Videos ;Container (6)
Command: select 1
Current Container - Music
Command: browse
1) [C:] Shared Music Container (84)
2) [C:] My Music ;Container (7)
3) Internet Radios ;Container (20)
Command: select 2
Current Container - [C:] My Music
Command: browse
1) Corel Sample Music ;Container (3)
2) iTunes ;Container (1)
3) Orchestra ;Container (2)
4) Get More with Jukebox Plus ;audio/mpeg
Command: select 3
Current Container - Orchestra
Command: browse
1) Anastasia ;Container (2)
Command: select 1
Current Container - Anastasia
Command: browse
1) Madeline5 ;audio/mpeg
2) 11 - Orchestra - Prologue ;audio/mpeg
Command: select 2
Current Item - 11 - Orchestra - Prologue
Command: metadata
MetaData command
Title: 11 - Orchestra - Prologue
Players: Orchestra
Class: object.item.audioItem.musicTrack
Uri: http://192.168.1.128:9500/
UPNPAV564480515U/21/C21/live.mp3
Protocol: http-get:*:audio/mpeg:DLNA.
ORG_PN=MP3;DLNA.ORG_
OP=01;DLNA.ORG_CI=0
Duration: 00:06:22.000
Command:
```

Command

mute

Description

Sets mute control for currently selected Media Renderer.

Syntax

mute <n>

Inputs:

n - where n is equal to

1 - mute on

0 - mute off

Outputs:

Echos the words "Mute command"

Example:

```
Command:          mute 1
Mute command
Command:          mute 0
Mute command
Command:
```


Command

name

Description

Define the friendly name for the RDMC 101 that is reported during SSDP. This will be the name of the device that appears in “My Network” that is associated with the RDMC 101. This is especially useful when more than one RDMC 101 unit is in use on a local network. After the friendly name has been changed, rebooting the RDMC 101 ensures the new name is broadcast. The RDMC 101 should be rebooted before issuing another `name` command.

Syntax

name “<name>”

Inputs:

name - where name is enclosed in double quotes if it contains spaces or other special characters. The max length of a name is 64 characters.

Outputs:

None.

Example:

```
Command: name "Allegro AV Demon RDMC
          Unit 1"
```

name

Command

next

Description

Select the current Server/Renderer/Container/Object + 1

Syntax

next

Inputs:

None.

Outputs:

None.

Example:

Command:

next

Command:

next

Command

nexttrack

Description

This command instructs the renderer to advance to the next track. This command is allowed when the renderer is playing or stopped.

Syntax

nexttrack

Inputs:

None.

Outputs:

None.

Example:

```
Command: nexttrack  
Command:
```

Command

pause

Description

Pause the current selected media stream

Syntax

pause

Inputs:

None.

Outputs:

None.

Example:

```
Command:          pause
Command:
```

pause

Command

play

Description

Commence sending current selected object from the selected server to the selected renderer.

Syntax

```
play <res>
```

Inputs:

`res` - Optional parameter - if more than one resource is available for the currently selected object the `play <res>` will send the selected resource to the renderer. In the case of audio, a selected song maybe stored on the media server in multiple formats (LPCM, MP3, AAC, etc.) or the media server might have the ability to transcode a stored object to another format while it is being served in realtime.

Outputs:

Displays the current Server, Renderer, Container and Object to the Telnet window.

Example:

```
Command: census
Census complete
Command: find
1) Orb: DB6TXNB1 - lshade ;Server
2) LAPTOP: Loren Shade: ;Server
3) SoundBridge ;Renderer
Command: select 2
Current Server - LAPTOP: Loren Shade:
Command: select 3
Current Renderer - SoundBridge
Command: browse
1) Music ;Container (10)
2) Video ;Container (7)
3) Pictures ;Container (7)
4) Playlists ;Container (2)
Command: select 1
Current Container - Music
Command: browse
1) All Music ;Container (1417)
2) Genre ;Container (23)
3) All Artists ;Container (115)
4) Contributing Artists ;Container (101)
5) Album Artists ;Container (107)
6) Composers ;Container (492)
7) Albums ;Container (157)
8) Rating ;Container (5)
9) Music Playlists ;Container (0)
10) Folders ;Container (2)
Command: select 1
Current Container - All Music
Command: browse
1) "Close Call" with Asteroid ;audio/mpeg
2) "Highway Blues" ;audio/x-ms-wma
3) 01 Track 1 ;audio/x-ms-wma
4) 1-2-3 ;audio/mpeg
5) 5 Reasons Businesses
   Must Embrace Podcasting
   - Marketing Online ;audio/mpeg
6) The 12 Days Of
   Christmas [Live] ;audio/mpeg
7) 17th Century Canon ;audio/mpeg
8) 19 Somethin' ;audio/mpeg
9) 25 or 6 to 4 ;audio/mpeg
10) 52nd Street Theme ;audio/mpeg
11) 100 Greatest Discoveries
    - Earth Sciences ;audio/mpeg
12) 100 Years ;audio/mpeg
```

```

13) About You
    (With All Of My Heart
    Album Version)           ;audio/mpeg
14) Abside the Winter       ;audio/mpeg
15) Accidentally In Love   ;audio/mpeg
16) Across the Lines       ;audio/mpeg
17) Across the Midnight Sky ;audio/mpeg
18) Across the Stars
    (Love Theme from Attack
    of the Clones)         ;audio/mpeg
19) An Actor's Life        ;audio/mpeg
20) Adventures [Unaltered] ;audio/mpeg
21) After Midnight        ;audio/mpeg
22) After Ventus          ;audio/mpeg
23) After We Make Love    ;audio/mpeg
24) After You've Gone     ;audio/mpeg
    Enter <browse> for more items
Command:                    select 3
Current Item - 01 Track 1
Command:                    play
Current Server - LAPTOP:    Loren Shade:
Current Renderer - SoundBridge
Current Container - All Music
Current Item - 01 Track 1
Command:

```

Command

playall

Description

Play all items in the current Container. To terminate the playall function enter any of the following: stop, up, find, browse or census.

Syntax

playall

Inputs:

None.

Outputs:

Displays the list of commands that will terminate the playall function. When Objects from the current Container are sent from the media server to the media renderer, the display is updated with the object name.

Example:

```
Command: census
Census complete
Command: find
1) Orb: DB6TXNB1 - lshade ;Server
2) SoundBridge ;Renderer
3) LAPTOP: Loren Shade: ;Server
Command: select 1
Current Server - Orb: DB6TXNB1 - lshade
Command: select 2
Current Renderer - SoundBridge
Command: browse
1) Music ;Container (3)
2) Pictures ;Container (2)
3) Videos ;Container (6)
Command: select 1
Current Container - Music
Command: browse
1) [C:] Shared Music ;Container (84)
2) [C:] My Music ;Container (7)
Command: select 2
Current Container - [C:] My Music
Command: browse
1) Corel Sample Music ;Container (3)
2) iTunes ;Container (1)
3) Orchestra ;Container (2)
4) Get More with Jukebox Plus ;audio/mpeg
Command: select 1
Current Container - Corel Sample Music
Command: browse
1) Classical Interlude 1 ;audio/mpeg
2) Jazz Groove ;audio/mpeg
3) Piano Blues 1 ;audio/mpeg
Command: playall
Enter stop, up, find, browse, or census to terminate
playall
Command:
Playing: Classical Interlude 1
Playing: Jazz Groove
Playing: Piano Blues 1
Playall Done
Command:
```

Command

playmode

Description

Set the playmode option

Syntax

playmode <n>

Inputs:

n - The argument supplied is a number that corresponds to one of the defined play modes. DIRECT_1 mode will play a single track and then stop. INTRO mode will play a short beginning of a track and then stop.

The argument should be a number from 0 to 6.

0 = NORMAL

1 = SHUFFLE

2 = REPEAT_ONE

3 = REPEAT_ALL

4 = RANDOM

5 = DIRECT_1

6 = INTRO

Outputs:

None.

Example:

Command

playnext

Description

Play selected item next on media renderer.

Syntax

playnext

Inputs:

None

Outputs:

Displays the media server and media renderer along with the Container and Object to be played next by the media renderer.

Example:

```
Command: census
Census complete
Command: find
1) SoundBridge ;Renderer
2) LAPTOP: Loren Shade: ;Server
3) Orb: DB6TXNB1 - lshade ;Server
Command: select 1
Current Renderer - SoundBridge
Command: select 3
Current Server - Orb: DB6TXNB1 - lshade
Command: browse
1) Music ;Container (3)
2) Pictures ;Container (2)
3) Videos ;Container (6)
Command: select 1
Current Container - Music
Command: browse
1) [C:] Shared Music ;Container (84)
2) [C:] My Music ;Container (7)
3) Internet Radios ;Container (20)
Command: select 2
Current Container - [C:] My Music
Command: browse
1) Corel Sample Music ;Container (3)
2) iTunes ;Container (1)
3) Orchestra ;Container (2)
4) Get More with Jukebox Plus ;audio/mpeg
Command: select 1
Current Container - Corel Sample Music
Command: browse
1) Classical Interlude 1 ;audio/mpeg
2) Jazz Groove ;audio/mpeg
3) Piano Blues 1 ;audio/mpeg
Command: select 1
Current Item - Classical Interlude 1
Command: play
Current Server - Orb: DB6TXNB1 - lshade
Current Renderer - SoundBridge
Current Container - Corel Sample Music
Current Item - Classical Interlude 1
Command: select 3
Current Item - Piano Blues 1
Command: playnext
Current Server - Orb: DB6TXNB1 - lshade
Current Renderer - SoundBridge
Current Container - Corel Sample Music
Current Item - Piano Blues 1
Command:
```

Command

previous

Description

Select the current Server/Renderer/Container/Object - 1

Syntax

previous

Inputs:

None.

Outputs:

Displays the name of the current Object.

Example:

```
Command: census
Census complete
Command: find
1) LAPTOP: Loren Shade: ;Server
2) SoundBridge ;Renderer
3) Orb: DB6TXNB1 - lshade ;Server
Command: select 2
Current Renderer - SoundBridge
Command: select 3
Current Server - Orb: DB6TXNB1 - lshade
Command: browse
1) Music ;Container (3)
2) Pictures ;Container (2)
3) Videos ;Container (6)
Command: select 1
Current Container - Music
Command: browse
1) [C:] Shared Music ;Container (84)
2) [C:] My Music ;Container (7)
3) Internet Radios ;Container (20)
Command: select 2
Current Container - [C:] My Music
Command: browse
1) Corel Sample Music ;Container (3)
2) iTunes ;Container (1)
3) Orchestra ;Container (2)
4) Get More with Jukebox Plus ;audio/mpeg
Command: select 1
Current Container - Corel Sample Music
Command: browse
1) Classical Interlude 1 ;audio/mpeg
2) Jazz Groove ;audio/mpeg
3) Piano Blues 1 ;audio/mpeg
Command: select 3
Current Item - Piano Blues 1
Command: play
Current Server - Orb: DB6TXNB1 - lshade
Current Renderer - SoundBridge
Current Container - Corel Sample Music
Current Item - Piano Blues 1
Command: previous
Current Item - Jazz Groove
Command: play
Current Server - Orb: DB6TXNB1 - lshade
Current Renderer - SoundBridge
Current Container - Corel Sample Music
Current Item - Jazz Groove
Command:
```

Command

previoustrack

Description

This command instructs the renderer to advance to the previous track. This command is allowed when the renderer is playing or stopped.

Syntax

previoustrack

Inputs:

None.

Outputs:

Echos the command.

Example:

```
Command: census
Census complete
Command: find
1) Orb: DB6TXNB1 - lshade ;Server
2) SoundBridge ;Renderer
3) LAPTOP: Loren Shade: ;Server
Command: select 2
Current Renderer - SoundBridge
Command: select 3
Current Server - LAPTOP: Loren Shade:
Command: select 1
Current Server - Orb: DB6TXNB1 - lshade
Command: browse
1) Music ;Container (3)
2) Pictures ;Container (2)
3) Videos ;Container (6)
Command: select 1
Current Container - Music
Command: browse
1) [C:] Shared Music ;Container (84)
2) [C:] My Music ;Container (7)
3) Internet Radios ;Container (20)
Command: select 2
Current Container - [C:] My Music
Command: browse
1) Corel Sample Music ;Container (3)
2) iTunes ;Container (1)
3) Orchestra ;Container (2)
4) Get More with Jukebox Plus ;audio/mpeg
Command: select 1
Current Container - Corel Sample Music
Command: browse
1) Classical Interlude 1 ;audio/mpeg
2) Jazz Groove ;audio/mpeg
3) Piano Blues 1 ;audio/mpeg
Command: playall
Enter stop, up, find, browse, or census to terminate
playall
Command:
Playing: Classical Interlude 1
Playing: Jazz Groove
Playing: Piano Blues 1
Playall Done
Command: previoustrack
PreviousTrack command
```



```
Command:                play
Current Server - Orb:    DB6TXNB1 - 1shade
Current Renderer - SoundBridge
Current Container - Corel Sample Music
Current Item - Piano Blues 1
Command:
```

Command

quit

Description

Quit the Telnet session and closes the connection

Syntax

quit

Inputs:

None.

Outputs:

None.

Example:

Command: quit

quit

Command

record

Description

Start record. Note this is only valid on renderers that can record.

Syntax

record

Inputs:

None.

Outputs:

None.

Example:

Command: record

Command

recordquality

Description

Set the record quality. Note this function must be implemented on the Renderer.

Syntax

recordquality

Inputs:

None.

Outputs:

None.

Example:

Command: recordquality

Command

resume

Description

Resume a paused media stream

Syntax

resume

Inputs:

None.

Outputs:

None.

Example:

```
Command: census
Census complete
Command: find
1) Orb: DB6TXNB1 - lshade ;Server
2) SoundBridge ;Renderer
3) LAPTOP: Loren Shade: ;Server
Command: select 2
Current Renderer - SoundBridge
Command: select 1
Current Server - Orb: DB6TXNB1 - lshade
Command: browse
1) Music ;Container (3)
2) Pictures ;Container (2)
3) Videos ;Container (6)
Command: select 1
Current Container - Music
Command: browse
1) [C:] Shared Music ;Container (84)
2) [C:] My Music ;Container (7)
3) Internet Radios ;Container (20)
Command: select 2
Current Container - [C:] My Music
Command: browse
1) Corel Sample Music ;Container (3)
2) iTunes ;Container (1)
3) Orchestra ;Container (2)
4) Get More with Jukebox Plus ;audio/mpeg
Command: select 1
Current Container - Corel Sample Music
Command: browse
1) Classical Interlude 1 ;audio/mpeg
2) Jazz Groove ;audio/mpeg
3) Piano Blues 1 ;audio/mpeg
Command: select 1
Current Item - Classical Interlude 1
Command: play
Current Server - Orb: DB6TXNB1 - lshade
Current Renderer - SoundBridge
Current Container - Corel Sample Music
Current Item - Classical Interlude 1
Command: pause
Renderer paused
Command: resume
Renderer resumed
Command:
```

Command

seek

Description

Position current item

Syntax

seek

Inputs:

Outputs:

None.

Example:

Command: seek

Command

select

Description

Select a Server/Renderer/Container/Object

Syntax

select

Inputs:

None.

Outputs:

None.

Example:

```
Command:                census
Census complete
Command:                find
1) Orb: DB6TXNB1 - lshade ;Server
2) SoundBridge          ;Renderer
3) LAPTOP: Loren Shade: ;Server
Command:                select 2
Current Renderer - SoundBridge
Command:                select 1
Current Server - Orb:   DB6TXNB1 - lshade
Command:                browse
1) Music                ;Container (3)
2) Pictures              ;Container (2)
3) Videos              ;Container (6)
Command:                select 1
Current Container - Music
Command:
```


Command

show

Description

Show current selections

Syntax

show

Inputs:

None.

Outputs:

Displays current selected Server, Renderer, Container and Object.

Example:

```
Command: census
Census complete
Command: find
1) Orb: DB6TXNB1 - lshade ;Server
2) LAPTOP: Loren Shade: ;Server
3) SoundBridge ;Renderer
Command: select 3
Current Renderer - SoundBridge
Command: select 1
Current Server - Orb: DB6TXNB1 - lshade
Command: browse
1) Music ;Container (3)
2) Pictures ;Container (2)
3) Videos ;Container (6)
Command: select 1
Current Container - Music
Command: browse
1) [C:] Shared Music ;Container (84)
2) [C:] My Music ;Container (7)
3) Internet Radios ;Container (20)
Command: select 2
Current Container - [C:] My Music
Command: browse
1) Corel Sample Music ;Container (3)
2) iTunes ;Container (1)
3) Orchestra ;Container (2)
4) Get More with Jukebox Plus ;audio/mpeg
Command: select 1
Current Container - Corel Sample Music
Command: browse
1) Classical Interlude 1 ;audio/mpeg
2) Jazz Groove ;audio/mpeg
3) Piano Blues 1 ;audio/mpeg
Command: select 2
Current Item - Jazz Groove
Command: show
Current Server - Orb: DB6TXNB1 - lshade
Current Renderer - SoundBridge
Current Container - Music
Current Container - [C:] My Music
Current Container - Corel Sample Music
Current Item - Jazz Groove
Command:
```

Command

status

Description

Display status for current renderer and transport

Syntax

status

Inputs:

None.

Outputs:

Displays a detailed list of transport attributes and their values.

Example:

```
Command:                census
Census complete
Command:                find
1) Orb: DB6TXNB1 - lshade ;Server
2) LAPTOP: Loren Shade: ;Server
3) SoundBridge          ;Renderer
Command:                select 3
Current Renderer - SoundBridge
Command:                select 1
Current Server - Orb:   DB6TXNB1 - lshade
Command:                status
Play Speeds - 1
Play Modes - NORMAL,SHUFFLE,REPEAT_ONE,REPEAT_ALL
Max Volume - 100
Optional Actions - Pause,SetNextAVTransportURI,
SetPlayMode,GetCurrentTransportActions
Transport State - PLAYING
Transport Status - OK
Transport Speed - 1
Position Title - Jonathan FritzÃ©n - Temple of Dreams -
Love Birds
Position Duration - 0:00:00
Position Time - 0:07:50
Track - 11
Media CurrentURI - unknown track uri
Media NextURI - unknown next transport uri
Media Medium - NETWORK
Number of Tracks - 18
Mute Value - false
Command:
```

Command

stop

Description

Stop the current media stream

Syntax

stop

Inputs:

None.

Outputs:

Displays - Renderer stopped

stop

Example:

```
Command: census
Census complete
Command: find
1) LAPTOP: Loren Shade: ;Server
2) SoundBridge ;Renderer
3) Orb: DB6TXNB1 - lshade ;Server
Command: select 2
Current Renderer - SoundBridge
Command: select 3
Current Server - Orb: DB6TXNB1 - lshade
Command: browse
1) Music ;Container (3)
2) Pictures ;Container (2)
3) Videos ;Container (6)
Command: select 1
Current Container - Music
Command: browse
1) [C:] Shared Music ;Container (84)
2) [C:] My Music ;Container (7)
3) Internet Radios ;Container (20)
Command: select 2
Current Container - [C:] My Music
Command: browse
1) Corel Sample Music ;Container (3)
2) iTunes ;Container (1)
3) Orchestra ;Container (2)
4) Get More with Jukebox Plus ;audio/mpeg
Command: select 1
Current Container - Corel Sample Music
Command: browse
1) Classical Interlude 1 ;audio/mpeg
2) Jazz Groove ;audio/mpeg
3) Piano Blues 1 ;audio/mpeg
Command: select 2
Current Item - Jazz Groove
Command: play
Current Server - Orb: DB6TXNB1 - lshade
Current Renderer - SoundBridge
Current Container - Corel Sample Music
Current Item - Jazz Groove
Command: stop
Renderer stopped
Command:
```

stop

Command

subscribe

Description

Subscribe to a Renderer AVTransport events

Syntax

subscribe

Inputs:

None.

Outputs:

Displays Renderer AVTransport events. Use the unsubscribe command to stop display of AVTransport events.

Example:

```
Command:                census
Census complete
Command:                find
1) LAPTOP: Loren Shade: ;Server
2) SoundBridge          ;Renderer
3) Orb: DB6TXNB1 - lshade ;Server
Command:                select 2
Current Renderer - SoundBridge
Command:                subscribe
Subscription complete
Command:
Event:                  PLAYING
Event:                  "Victor Fields - This Could Be
                        Paradise - Victor"
Event:                  PAUSED_PLAYBACK
Event:                  PLAYING
Event:                  "Smoothjazz.com (MP3 128k)"
Event:                  "Victor Fields - This Could Be
                        Paradise - Victor"
Event:                  "Everette Harp -
                        All Jazzed Up - My Inspiration"
Command:                unsubscribe
Unsubscription complete
Command:
```


Command

transportactions

Description

Display current transport actions.

Syntax

transportactions

Inputs:

None.

Outputs:

Displays a list of transport actions supported by the Renderer.

Example:

```

Command:                census
Census complete
Command:                find
1) LAPTOP: Loren   Shade:   ;Server
2) SoundBridge    ;Renderer
3) Orb: DB6TXNB1 - lshade  ;Server
Command:                select 2
Current Renderer - SoundBridge
Command:                transportactions
Current Transport Actions command
Actions:                Play,Stop,Pause
Command:

```

Command

unsubscribe

Description

Unsubscribe from the Renderer AVTransport actions

Syntax

unsubscribe

Inputs:

None.

Outputs:

Displays the message Unsubscription complete.

Example:

```
Command:                census
Census complete
Command:                find
1) LAPTOP: Loren Shade: ;Server
2) SoundBridge          ;Renderer
3) Orb: DB6TXNB1 - lshade ;Server
Command:                select 2
Current Renderer - SoundBridge
Command:                subscribe
Subscription complete
Command:
Event:                  PLAYING
Event: "                Victor Fields -
                        This Could Be Paradise - Victor"
Event:                  PAUSED_PLAYBACK
Event:                  PLAYING
Event:                  "Smoothjazz.com (MP3 128k)"
Event:                  "Victor Fields -
                        This Could Be Paradise - Victor"
Event:                  "Everette Harp -
                        All Jazzed Up - My Inspiration"
Command:                unsubscribe
Unsubscription complete
Command:
```

unsubscribe

Command

up

Description

Select parent Container of the current Container/Object.

Syntax

up

Inputs:

None.

Outputs:

Displays the current Containter/Renderer.

Example:

```
Command: census
Census complete
Command: find
1) LAPTOP: Loren Shade: ;Server
2) SoundBridge ;Renderer
3) Orb: DB6TXNB1 - lshade ;Server
Command: select 1
Current Server - LAPTOP: Loren Shade:
Command: browse
1) Music ;Container (10)
2) Video ;Container (7)
3) Pictures ;Container (7)
4) Playlists ;Container (2)
Command: select 1
Current Container - Music
Command: browse
1) All Music ;Container (1417)
2) Genre ;Container (23)
3) All Artists ;Container (115)
Command: select 1
Current Container - All Music
Command: browse
1) "Close Call" with Asteroid ;audio/mpeg
2) "Highway Blues" ;audio/x-ms-wma
3) 01 Track 1 ;audio/x-ms-wma
4) 1-2-3 ;audio/mpeg
5) 5 Reasons Businesses
Must Embrace Podcasting -
Marketing Online ;audio/mpeg
6) The 12 Days Of
Christmas [Live] ;audio/mpeg
7) 17th Century Canon ;audio/mpeg
8) 19 Somethin' ;audio/mpeg
9) 25 or 6 to 4 ;audio/mpeg
10) 52nd Street Theme ;audio/mpeg
11) 100 Greatest Discoveries
- Earth Sciences ;audio/mpeg
12) 100 Years ;audio/mpeg
13) About You (With All Of
My Heart Album Version) ;audio/mpeg
14) Abside the Winter ;audio/mpeg
15) Accidentally In Love ;audio/mpeg
16) Across the Lines ;audio/mpeg
17) Across the Midnight Sky ;audio/mpeg
18) Across the Stars
(Love Theme from Attack of
the Clones) ;audio/mpeg
19) An Actor's Life ;audio/mpeg
20) Adventures [Unaltered] ;audio/mpeg
```

```
21) After Midnight           ;audio/mpeg
22) After Ventus             ;audio/mpeg
23) After We Make Love      ;audio/mpeg
24) After You've Gone       ;audio/mpeg
Enter <browse> for more items
Command:                     up
Current Container - Music
Command:                     up
Current Selection - LAPTOP: Loren Shade:
Command:                     browse
1) Music                     ;Container (10)
2) Video                     ;Container (7)
3) Pictures                  ;Container (7)
4) Playlists                 ;Container (2)
Command:
```

Command

upload

Description

Upload a file to a Server.

Syntax

upload

Inputs:

Outputs:

Displays the current Containter/Renderer.

Example:

Command: census

Command

volume

Description

Upload a file to a Server.

Syntax

volume <n>

Inputs:

A number between 0 - 100 representing the volume.

Outputs:

Displays current volume setting.

Example:

```

Command:                census
Census complete
Command:                find
1) LAPTOP: Loren Shade: ;Server
2) Allegro Media Server ;Server
3) SoundBridge          ;Renderer
4) Orb: DB6TXNB1 - lshade ;Server
Command:                select 3
Current Renderer - SoundBridge
Command:                volume 45
Current Volume - 45
Command:
    
```