MediaServer:1 Device Template Version 1.01

For UPnP™ Version 1.0
Status: Standardized DCP
Date: June 25, 2002

THE UPNP™ FORUM TAKES NO POSITION AS TO WHETHER ANY INTELLECTUAL PROPERTY RIGHTS EXIST IN THE STANDARDIZED DCPS. THE STANDARDIZED DCPS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". THE UPNP™ FORUM MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE STANDARDIZED DCPS, INCLUDING BUT NOT LIMITED TO ALL IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OF REASONABLE CARE OR WORKMANLIKE EFFORT, OR RESULTS OR OF LACK OF NEGLIGENCE.

Authors	Company
John Ritchie	Intel

Contents

List of Tables

1. Overview and Scope

This device template is compliant with the UPnP™ Device Architecture version 1.0. It defines a device type referred to herein as MediaServer:1.

The MediaServer template defines a general-purpose device that can be used to instantiate any Consumer Electronic (CE) device that provides AV content (e.g. media) to other UPnP devices on the home network. It is based on the UPnP AV Architecture Framework (described in another document). It exposes its content via the Content Directory service (refer to the Content Directory Service Template for details). As such, the MediaServer can handle any specific type of media, any data format, and transfer protocol.

Example instances of a MediaServer include traditional devices such as VCRs, CD Players, DVD Players, audio-tape players, still-image cameras, camcorders, radios, TV Tuners, and set-top boxes. Additional examples of a MediaServer also include new digital devices such as MP3 servers, PVRs, and Home Media Servers such as the PC. All though these devices contain diverse (AV) content in one form or another, the MediaServer (via the Content Directory) is able to expose this content to the home network in a uniform and consistent manner. This ability allows the MediaServer to instantiate traditional single-function devices as well as more recent multi-function devices such as VCR-DVD players and the general purpose Home Media Server, which contains a wide-variety of content such as MPEG2 video, CD audio, MP3 and/or WMA audio, JPEG images, etc.

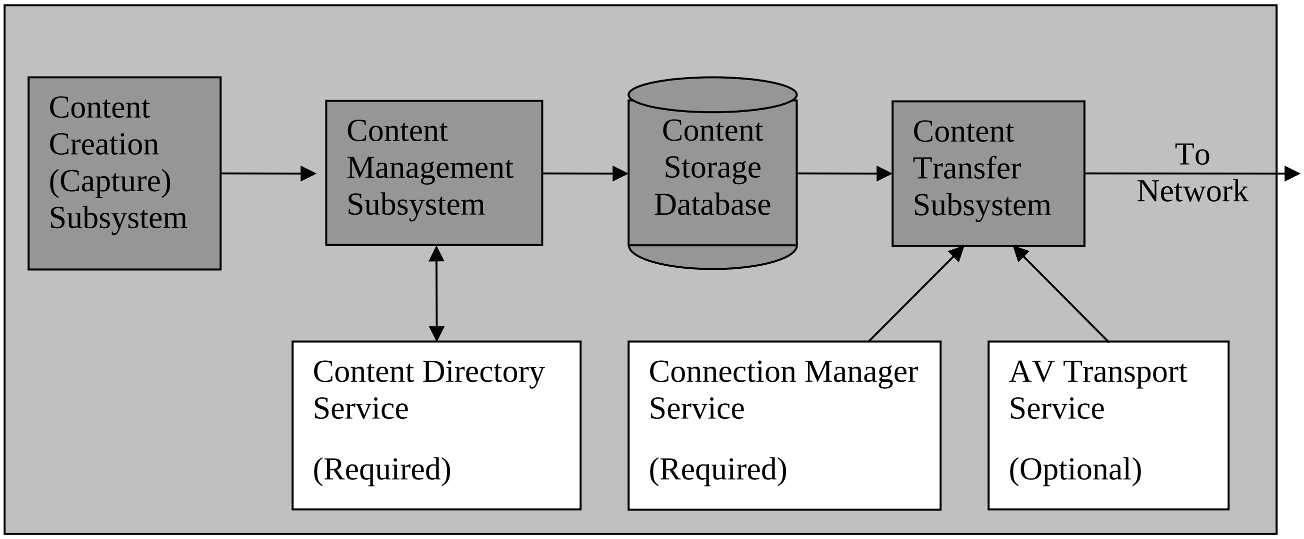
The MediaServer template is very lightweight and can easily be implemented on low-resource devices such as still-image cameras or MP3 players that want to expose their local content to the home network. The MediaServer can also be used for high-end Home Media Servers that contain dozens of Gigabytes of heterogeneous content. Refer to the Theory Of Operation section for some specific examples of the MediaServer.

A full-featured MediaServer device provides clients with the following capabilities:

- Enumerate and query any of the content that the MediaServer can provide to the home network.
- Negotiate a common transfer protocol and data format between the MediaServer and target device.
- Control the flow of the content (e.g. FF, REW, etc).
- Copy (import) content to the MediaServer from another device.

This device template does not provide:

- The ability to renderer AV content.


Content Creation (Capture) Subsystem
Content Management Subsystem
Content Storage Database
Content Transfer Subsystem
To Network
Content Directory Service
(Required)
Connection Manager Service
(Required)
AV Transport Service
(Optional)

MediaServer Device

Figure 1: MediaServer:1 Functional Diagram – The un-shaded blocks represent the UPnP services that are contained by a MediaServer device. The shaded blocks represent various device-specific modules that the UPnP services might interact with. However, the internal architecture of a MediaServer device is vendor specific.

2. Device Definitions

2.1. Device Type

The following device type identifies a device that is compliant with this template:

urn:schemas-upnp-org:device:MediaServer:1

The shorthand MediaServer:1 is used herein to refer to this device type.

2.2. Device Model

MediaServer:1 products must implement minimum version numbers of all required embedded devices and services specified in the table below.

Table 1: Device Requirements

DeviceType	Root	Req. or Opt.[1]	ServiceType	Req. or Opt.[1]	Service ID[2]
MediaServer:1	YES	R	ContentDirectory:1.0	R	ContentDirectory
			ConnectionManager:1.0	R	ConnectorManager
			AVTransport:1.0	O	AVTransport
			Non-standard services embedded by an UPnP vendor go here.	*X*	*TBD*
Non-standard devices embedded by a UPnP vendor go here.	*TBD*	*X*	*TBD*	*TBD*	*TBD*

[1] R = Required, O = Optional, X = Non-standard.
[2] Prefixed by urn:**upnp-org**:**serviceId**: .

2.2.1. Description of Device Requirements

Each implementation of the MediaServer requires a Content Directory and ConnectionManager service. The Content Directory service allows Control Points to discover information about the AV content that is available from the device. The Connection Manager is used to enumerate and select a particular transfer protocol and data format to be used for transferring the content. Additionally, the Connection Manager also allows Control Points, such as a home network management application, to discover useful information about the content transfers that the device is actively participating in. Such information could be useful to a Quality Of Service capability, which may be defined in the future..

The existence of the AVTransport service depends on the transfer protocols that are supported the device. The Connection Manager specification includes a table that identifies which transfer protocols require an AVTransport to be implemented on the MediaServer. If an implementation of the MediaServer supports any of these transfer protocols, then it must implement the AVTransport service. However, no AVTransport instances will be instantiated until a connection is made using one of those transfer protocol.

2.2.2. Relationships Between Services

The Connection Manager's PrepareForConnection() action provides the trigger point for creating new instances of the AVTransport service. When a new connection is established (one that requires an instance of the AVTransport on the MediaServer, which is determined by the selected transfer protocol), the PrepareForConnection() action returns the InstanceId of the AVTransport that is bound to that connection. This instance is used by the Control Point to control the flow (e.g. Play, FF, REW, Seek, etc) of the content to the network. As described in the AVTransport specification, each instance of the AVTransport service operates independently.

2.3. Theory of Operation

MediaServer devices are used in conjunction with one or more MediaRenderer device(s) to allow a Control Point to discover entertainment (AV) content (e.g. video, music, images, etc) on the MediaServer and to render that content on any appropriate MediaRenderer within the home network. In general terms, the process begins with the Control Points discovering MediaServer and MediaRenderer devices within the home network. The Control Point interacts with a MediaServer(s) to locate a desired piece of content (e.g. a movie, a song, a playlist, a photo album, etc). After the content has been identified, the Control point needs to identify a common transfer protocol and data format that can be used to transfer the content from the MediaServer to the desired MediaRenderer. After these transfer parameters have been established, the Control Point controls the flow of the content (e.g. Play, Pause, Stop, Seek, etc.). (Depending on the selected transfer protocol, these flow control operations are sent either to the MediaServer or MediaRenderer, but not both). The actual transfer of the content is performed directly by the MediaServer and MediaRenderer. The content transfer happens independently from the Control Point and does not involve UPnP itself at all. The Control Point uses UPnP to setup the transfer of the content, but the transfer is performed using a transfer protocol other than UPnP.

2.3.1. Device Discovery:

Control Points can discover MediaServer devices using the standard UPnP SSDP-based device discovery mechanism to search for any device that is a “MediaServer” device class including root devices and/or embedded devices.

2.3.2. Locating Desired Content:

Control Points use the MediaServer’s ContentDirectory service to locate desired content. The ContentDirectory service exposes both a search capability and a browse capability. Searching is useful when the Control Point (via the end-user) knows something about the content it wants to find (e.g. its name, artist, type, date created, etc). Browsing is useful for blindly discovering what content the device has to offer. Each content item that is referenced by the ContentDirectory service includes various information about that content including the transfer protocol(s) and file format(s) that the MediaServer can use to transfer the content to the MediaRenderer.

2.3.3. Preparing to Transfer the Content:

After the desired content has been identified, the Control Point needs to determine which transfer protocol and data format should be used to transfer the content from the MediaServer to the MediaRenderer. (Transfer protocol examples include IEEE-1394, HTTP GET, RTSP/RTP, etc., and data format examples include MPEG2, MPEG4, MP3, WMA, JPEG, etc.) The Control Point makes this determination by comparing the content’s protocol/format information (obtained via the MediaServer’s ContentDirectory Service) with the protocol/format information obtained via the MediaRenderer’s ConnectionManager::GetProtocolInfo() action.

After the transfer protocol and data format have been identified, the Control Point uses the ConnectionManager::PrepareForConnection() action on each device to inform the device that the specified protocol/format are about to be used. Depending on which transfer protocol was selected, the PrepareForConnection() action on either the MediaServer or MediaRenderer will return an AVTransport InstanceID to the Control Point. This AVTransport InstanceID is used by the Control Point to control the transfer of the content (e.g. Play, Pause, Stop, Seek, etc). Refer to the subsection below for more details.

Depending on which transfer protocols are supported by the device (e.g. devices that only support HTTP GET), a MediaServer and/or MediaRenderer may choose to NOT implement the PrepareForConnection() action. In this case, the Control Point may not have been able to obtain an AVTransport InstanceID from either device. When this happens, the Control Point uses an AVTransport InstanceID of 0 (zero). If the MediaRenderer has implemented the

AVTransport Service, the Control Point should use it for all AVTransport actions. Otherwise, AVTransport actions should be sent to the MediaServer device. Refer to the ConnectionManager Service for more information.

2.3.4. Controlling the Transfer of the Content:

Regardless of which device provided the AVTransport InstanceID, the Control Point uses the AVTranport Service on that device to control the flow of the content. For example, to begin transferring the content, the Control Point invokes the AVTransport::Play() action. To skip to a specific location within the content, the Control Point invokes the AVTransport::Seek() action. In most cases, the choice of AVTransport actions that are actually invoked will likely be directed by the end-user as they interact with the Control Point's UI. Refer to the AVTransport Service for additional details of these and other AVTransport actions.

3. XML Device Description

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <URLBase>base URL for all relative URLs</URLBase>
  <device>
    <deviceType>urn:schemas-upnp-org:device:MediaServer:1</deviceType>
    <friendlyName>short user-friendly title</friendlyName>
    <manufacturer>manufacturer name</manufacturer>
    <manufacturerURL>URL to manufacturer site</manufacturerURL>
    <modelDescription>long user-friendly title</modelDescription>
    <modelName>model name</modelName>
    <modelNumber>model number</modelNumber>
    <modelURL>URL to model site</modelURL>
    <serialNumber>manufacturer's serial number</serialNumber>
    <UDN>uuid:UUID</UDN>
    <UPC>Universal Product Code</UPC>
    <iconList>
      <icon>
        <mimetype>image/format</mimetype>
        <width>horizontal pixels</width>
        <height>vertical pixels</height>
        <depth>color depth</depth>
        <url>URL to icon</url>
      </icon>
      XML to declare other icons, if any, go here
    </iconList>
    <serviceList>
      <service>
        <serviceType>urn:schemas-upnp-org:service:ContentDirectory:1</serviceType>
        <serviceId>urn:upnp-org:serviceId:ContentDirectory</serviceId>
        <SCPDURL>URL to service description</SCPDURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
      </service>
      <service>
        <serviceType>urn:schemas-upnp-org:service:ConnectionManager:1</serviceType>
        <serviceId>urn:upnp-org:serviceId:ConnectionManager</serviceId>
        <SCPDURL>URL to service description</SCPDURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
      </service>
      <service>
        <serviceType>urn:schemas-upnp-org:service:AVTransport:1</serviceType>
        <serviceId>urn:upnp-org:serviceId:AVTransport</serviceId>
        <SCPDURL>URL to service description</SCPDURL>
        <controlURL>URL for control</controlURL>
        <eventSubURL>URL for eventing</eventSubURL>
      </service>
      Declarations for other services added by UPnP vendor (if any) go here
    </serviceList>
    <deviceList>
      Description of embedded devices added by UPnP vendor (if any) go here
    </deviceList>
    <presentationURL>URL for presentation</presentationURL>
  </device>
</root>
```

4. Test

There are no semantic tests defined for this device.