
MediaRenderer:1 Device Template Version 1.01

For UPnP™ 1.0

Status: Standardized DCP

Date: June 25, 2002

This Standardized DCP has been adopted as a Standardized DCP by the Steering Committee of the UPnP™ Forum, pursuant to Section 2.1(c)(ii) of the UPnP™ Forum Membership Agreement. UPnP™ Forum Members have rights and licenses defined by Section 3 of the UPnP™ Forum Membership Agreement to use and reproduce the Standardized DCP in UPnP™ Compliant Devices. All such use is subject to all of the provisions of the UPnP™ Forum Membership Agreement.

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.

© 1999-2002 Contributing Members of the UPnP™ Forum. All rights Reserved.

Authors		Company	
John Ritchie		Intel	

Contents

1. OVERVIEW AND SCOPE.....	3
2. DEVICE DEFINITIONS	5
2.1. DEVICE TYPE.....	5
2.2. DEVICE MODEL	5
2.2.1. <i>Description of Device Requirements</i>	5
2.2.2. <i>Relationships Between Services</i>	6
2.3. THEORY OF OPERATION.....	7
2.3.1. <i>Device Discovery:</i>	7
2.3.2. <i>Preparing to Transfer the Content:</i>	7
2.3.3. <i>Controlling the Transfer of the Content:</i>	8
2.3.4. <i>Controlling “How” the Content is Rendered.</i>	8
3. XML DEVICE DESCRIPTION.....	9
4. TEST.....	11

List of Tables

Table 1: Device Requirements	5
------------------------------------	---

1. Overview and Scope

This device template is compliant with the Universal Plug and Play Device Architecture version 1.0. It defines a device type referred to herein as MediaRenderer:1.

The MediaRenderer template defines a general-purpose device template that can be used to instantiate any Consumer Electronic (CE) device that is capable of rendering AV content from the home network. It exposes a set of rendering controls in which a Control Point can control how the specified AV content is rendered. This includes controlling various rendering features such as brightness, contrast, volume, etc.

Example instances of a MediaRenderer include traditional devices such as TVs and stereo systems. Some more contemporary examples include digital devices such as MP3 players and Electronic Picture Frames (EPFs). All though most of these examples typically render one specific type of content (e.g. a TV typically renders video content), a MediaRenderer is able to support a number of different data formats and transfer protocols. For example, a sophisticated implementation of a TV MediaRenderer could also support MP3 data so that its speakers could be used to play MP3 audio content.

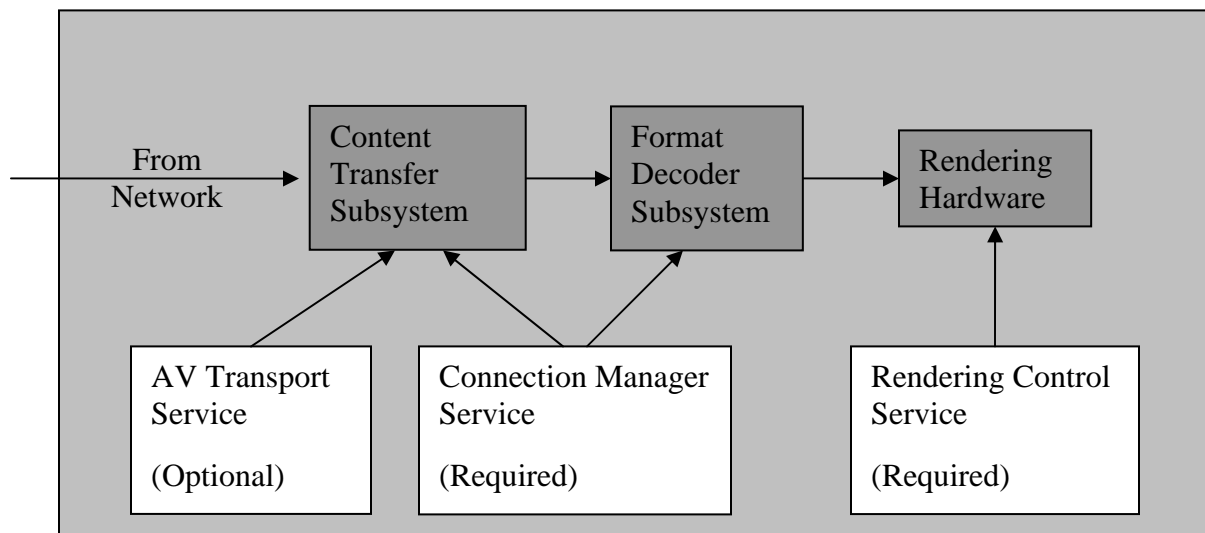
The MediaRenderer device template is very lightweight and is easy to implement on low-resource devices such as an MP3 player. However, it can also be used to expose the high-end capabilities of such device as a PC.

A full-featured MediaRenderer exposes the following capabilities:

- Control various rendering characteristics
- Expose the supported transfer protocols and data formats
- Control the flow of the content (e.g. FF, REW, etc), if appropriate depending on the transfer protocol.

The MediaRenderer DOES NOT enable Control Points to:

- Send AV content to another device
- Retrieve any type of meta-data associated with the content



MediaRenderer

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

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

2.2. Device Model

MediaRenderer: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
MediaRenderer:1	YES	R	RenderingControl:1.0	R	RenderingControl
			ConnectionManager:1.0	R	ConnectionManager
			AVTransport:1.0	O	AVTransport
			<i>Non-standard services embedded by an UPnP vendor go here.</i>	<i>X</i>	<i>TBD</i>
<i>Non-standard devices embedded by a UPnP vendor go here.</i>	<i>TBD</i>	<i>X</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>

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 MediaRenderer requires a Rendering Control and ConnectionManager service. The Rendering Control service allows Control Points to control the various rendering capabilities of 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 by the device. The Connection Manager specification includes a table that identifies which transfer protocols require an AVTransport to be implemented on the MediaRenderer. If an implementation of the MediaRenderer 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 Rendering Control and AVTransport service. When a new connection is established (one that requires an instance of the AVTransport on the MediaRenderer, which is determined by the selected transfer protocol), the PrepareForConnection() action returns the InstanceId of the Rendering Control and AVTransport services that are bound to that connection. The Rendering Control instance is used by the Control Point to control how the content from that connection is rendered. The AVTransport instance is used by the Control Point to control the flow (e.g. Play, FF, REW, Seek, etc) of the content received via that connection. As described in the RenderingControl and AVTransport specification, each instance of these services operates independently from all other instances.

2.3. Theory of Operation

MediaRenderer devices are used in conjunction with one or more MediaServer device(s) to allow a Control Point to render entertainment (AV) content (e.g. video, music, images, etc) that is discovered on a MediaServer device within the home network. In general terms, the process begins with the Control Points discovering MediaServer and MediaRenderer devices within the home network. After the Control Point locates the desired content on a MediaServer, 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 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. 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 MediaRenderer 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. 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 MediaRenderer or MediaServer 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 MediaRenderer and/or MediaServer 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 should use 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.

2.3.3. Controlling the Transfer of the Content:

Regardless of which device provides the AVTransport InstanceID, the Control Point uses the AVTransport 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 information about these and other AVTransport actions.

2.3.4. Controlling “How” the Content is Rendered.

Similar to the allocation of AVTransport InstanceIDs via the MediaRenderer's ConnectionManager::PrepareForConnection() action will also return a RenderingControl InstanceId. This InstanceID is used in conjunction with the Rendering Control Service to control “how” the content is to be rendered. For example, to change the loudness of the sound, the Control Point invokes the RenderingControl::SetVolume(). The Control Point passes in the Rendering Control InstanceID and the desired volume setting. To get the current brightness of the MediaRenderer's display, the Control Point invokes the GetBrightness() action. The InstanceID is passed in and the current brightness setting is returned.. Refer to the RenderingControl Service for additional details on these and other actions that affect how content is rendered.

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:MediaRenderer: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>
    <modelName>model number</modelName>
    <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:RenderingControl:1</serviceType>
        <serviceId>urn:upnp-org:serviceId:RenderingControl</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>
    </serviceList>
  </device>
</root>

```

```
</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 symantic tests defined for this device.

