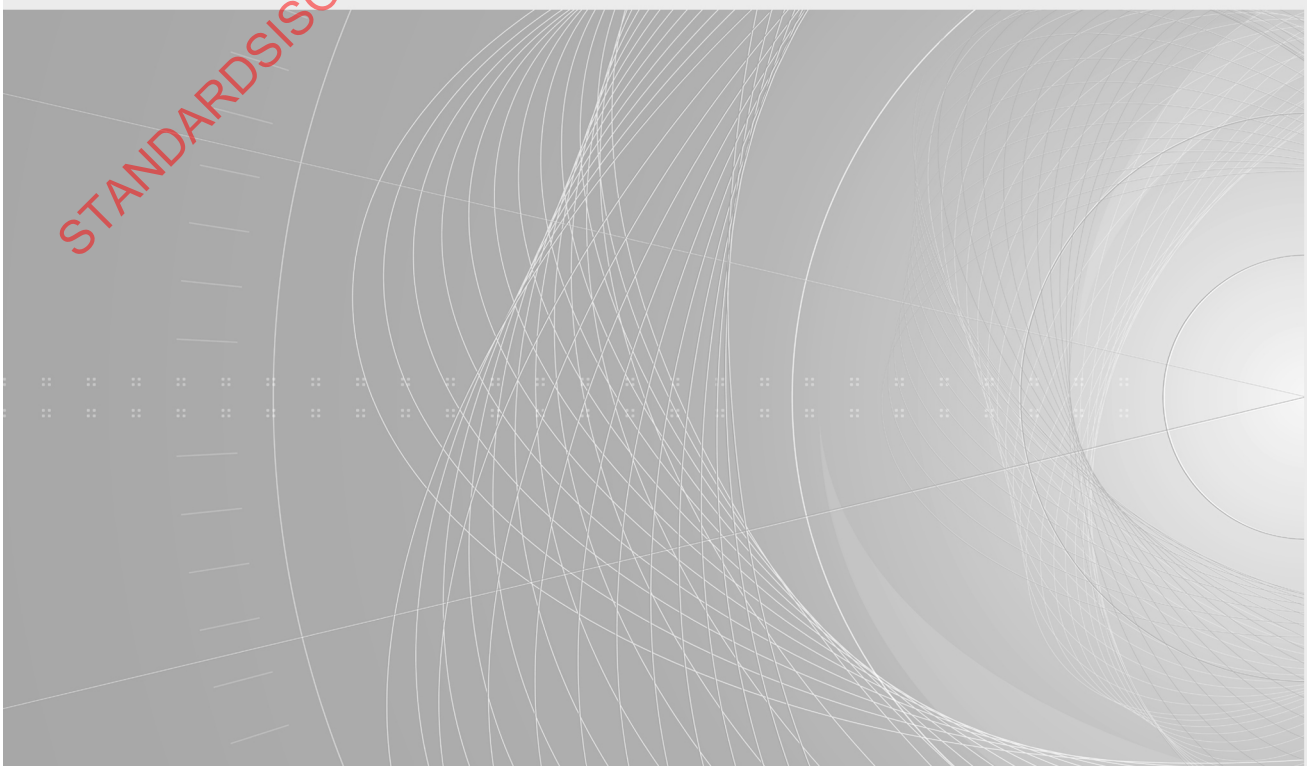
ISO/IEC 29341-3-13

Edition 1.0 2008-11

INTERNATIONAL STANDARD

Information technology – UPnP Device Architecture – Part 3-13: Audio Video Device Control Protocol – Rendering Control Service



ISO/IEC 29341-3-13:2008(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub
The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub
Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv
If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

ISO/IEC 29341-3-13

Edition 1.0 2008-11

INTERNATIONAL STANDARD

Information technology – UPnP Device Architecture – Part 3-13: Audio Video Device Control Protocol – Rendering Control Service

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PRICE CODE U

ICS 35.200

ISBN 2-8318-1005-7

CONTENTS

LIST OF TABLES

INFORMATION TECHNOLOGY – UPNP DEVICE ARCHITECTURE –

Part 3-13: Audio Video Device Control Protocol – Rendering Control Service

FOREWORD

1) ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

3) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO member bodies.

4) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

5) In order to promote international uniformity, IEC and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.

6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC publication.

7) All users should ensure that they have the latest edition of this publication.

8) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.

9) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

IEC and ISO draw attention to the fact that it is claimed that compliance with this document may involve the use of patents as indicated below.

ISO and IEC take no position concerning the evidence, validity and scope of the putative patent rights. The holders of the putative patent rights have assured IEC and ISO that they are willing to negotiate free licences or licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of the putative patent rights are registered with IEC and ISO.

Intel Corporation has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Intel Corporation
Standards Licensing Department
5200 NE Elam Young Parkway
MS: JFS-98
USA – Hillsboro, Oregon 97124

Microsoft Corporation has informed IEC and ISO that it has patent applications or granted patents as listed below:

6101499 / US; 6687755 / US; 6910068 / US; 7130895 / US; 6725281 / US; 7089307 / US; 7069312 / US; 10/783 524 /US

Information may be obtained from:

Microsoft Corporation
One Microsoft Way
USA – Redmond WA 98052

Philips International B.V. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Philips International B.V. – IP&S
High Tech campus, building 44 3A21
NL – 5656 Eindhoven

NXP B.V. (NL) has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

NXP B.V. (NL)
High Tech campus 60
NL – 5656 AG Eindhoven

Matsushita Electric Industrial Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Matsushita Electric Industrial Co. Ltd.
1-3-7 Shiromi, Chuoh-ku
JP – Osaka 540-6139

Hewlett Packard Company has informed IEC and ISO that it has patent applications or granted patents as listed below:

5 956 487 / US; 6 170 007 / US; 6 139 177 / US; 6 529 936 / US; 6 470 339 / US; 6 571 388 / US; 6 205 466 / US

Information may be obtained from:

Hewlett Packard Company
1501 Page Mill Road
USA – Palo Alto, CA 94304

Samsung Electronics Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Digital Media Business, Samsung Electronics Co. Ltd.
416 Maetan-3 Dong, Yeongtang-Gu,
KR – Suwon City 443-742

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. IEC and ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 29341-3-13 was prepared by UPnP Implementers Corporation and adopted, under the PAS procedure, by joint technical committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

The list of all currently available parts of the ISO/IEC 29341 series, under the general title *Universal plug and play (UPnP) architecture*, can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

ORIGINAL UPNP DOCUMENTS
(informative)

Reference may be made in this document to original UPnP documents. These references are retained in order to maintain consistency between the specifications as published by ISO/IEC and by UPnP Implementers Corporation. The following table indicates the original UPnP document titles and the corresponding part of ISO/IEC 29341:

UPnP Document Title	ISO/IEC 29341 Part
UPnP Device Architecture 1.0	ISO/IEC 29341-1
UPnP Basic:1 Device	ISO/IEC 29341-2
UPnP AV Architecture:1	ISO/IEC 29341-3-1
UPnP MediaRenderer:1 Device	ISO/IEC 29341-3-2
UPnP MediaServer:1 Device	ISO/IEC 29341-3-3
UPnP AVTransport:1 Service	ISO/IEC 29341-3-10
UPnP ConnectionManager:1 Service	ISO/IEC 29341-3-11
UPnP ContentDirectory:1 Service	ISO/IEC 29341-3-12
UPnP RenderingControl:1 Service	ISO/IEC 29341-3-13
UPnP MediaRenderer:2 Device	ISO/IEC 29341-4-2
UPnP MediaServer:2 Device	ISO/IEC 29341-4-3
UPnP AV Datastructure Template:1	ISO/IEC 29341-4-4
UPnP AVTransport:2 Service	ISO/IEC 29341-4-10
UPnP ConnectionManager:2 Service	ISO/IEC 29341-4-11
UPnP ContentDirectory:2 Service	ISO/IEC 29341-4-12
UPnP RenderingControl:2 Service	ISO/IEC 29341-4-13
UPnP ScheduledRecording:1	ISO/IEC 29341-4-14
UPnP DigitalSecurityCamera:1 Device	ISO/IEC 29341-5-1
UPnP DigitalSecurityCameraMotionImage:1 Service	ISO/IEC 29341-5-10
UPnP DigitalSecurityCameraSettings:1 Service	ISO/IEC 29341-5-11
UPnP DigitalSecurityCameraStillImage:1 Service	ISO/IEC 29341-5-12
UPnP HVAC_System:1 Device	ISO/IEC 29341-6-1
UPnP HVAC_ZoneThermostat:1 Device	ISO/IEC 29341-6-2
UPnP ControlValve:1 Service	ISO/IEC 29341-6-10
UPnP HVAC_FanOperatingMode:1 Service	ISO/IEC 29341-6-11
UPnP FanSpeed:1 Service	ISO/IEC 29341-6-12
UPnP HouseStatus:1 Service	ISO/IEC 29341-6-13
UPnP HVAC_SetpointSchedule:1 Service	ISO/IEC 29341-6-14
UPnP TemperatureSensor:1 Service	ISO/IEC 29341-6-15
UPnP TemperatureSetpoint:1 Service	ISO/IEC 29341-6-16
UPnP HVAC_UserOperatingMode:1 Service	ISO/IEC 29341-6-17
UPnP BinaryLight:1 Device	ISO/IEC 29341-7-1
UPnP DimmableLight:1 Device	ISO/IEC 29341-7-2
UPnP Dimming:1 Service	ISO/IEC 29341-7-10
UPnP SwitchPower:1 Service	ISO/IEC 29341-7-11
UPnP InternetGatewayDevice:1 Device	ISO/IEC 29341-8-1
UPnP LANDevice:1 Device	ISO/IEC 29341-8-2
UPnP WANDevice:1 Device	ISO/IEC 29341-8-3
UPnP WANConnectionDevice:1 Device	ISO/IEC 29341-8-4
UPnP WLANAccessPointDevice:1 Device	ISO/IEC 29341-8-5
UPnP LANHostConfigManagement:1 Service	ISO/IEC 29341-8-10
UPnP Layer3Forwarding:1 Service	ISO/IEC 29341-8-11
UPnP LinkAuthentication:1 Service	ISO/IEC 29341-8-12
UPnP RadiusClient:1 Service	ISO/IEC 29341-8-13
UPnP WANCableLinkConfig:1 Service	ISO/IEC 29341-8-14
UPnP WANCommonInterfaceConfig:1 Service	ISO/IEC 29341-8-15
UPnP WANDSLLinkConfig:1 Service	ISO/IEC 29341-8-16
UPnP WANEthernetLinkConfig:1 Service	ISO/IEC 29341-8-17
UPnP WANIPConnection:1 Service	ISO/IEC 29341-8-18
UPnP WANPOTSLinkConfig:1 Service	ISO/IEC 29341-8-19
UPnP WANPPPConnection:1 Service	ISO/IEC 29341-8-20
UPnP WLANConfiguration:1 Service	ISO/IEC 29341-8-21
UPnP Printer:1 Device	ISO/IEC 29341-9-1
UPnP Scanner:1.0 Device	ISO/IEC 29341-9-2
UPnP ExternalActivity:1 Service	ISO/IEC 29341-9-10
UPnP Feeder:1.0 Service	ISO/IEC 29341-9-11
UPnP PrintBasic:1 Service	ISO/IEC 29341-9-12
UPnP Scan:1 Service	ISO/IEC 29341-9-13
UPnP QoS Architecture:1.0	ISO/IEC 29341-10-1
UPnP QosDevice:1 Service	ISO/IEC 29341-10-10
UPnP QosManager:1 Service	ISO/IEC 29341-10-11
UPnP QosPolicyHolder:1 Service	ISO/IEC 29341-10-12
UPnP QoS Architecture:2	ISO/IEC 29341-11-1
UPnP QOS v2 Schema Files	ISO/IEC 29341-11-2
UPnP QosDevice:2 Service	ISO/IEC 29341-11-10

UPnP Document Title	ISO/IEC 29341 Part
UPnP QosManager:2 Service	ISO/IEC 29341-11-11
UPnP QosPolicyHolder:2 Service	ISO/IEC 29341-11-12
UPnP RemoteUIClientDevice:1 Device	ISO/IEC 29341-12-1
UPnP RemoteUIServerDevice:1 Device	ISO/IEC 29341-12-2
UPnP RemoteUIClient:1 Service	ISO/IEC 29341-12-10
UPnP RemoteUIServer:1 Service	ISO/IEC 29341-12-11
UPnP DeviceSecurity:1 Service	ISO/IEC 29341-13-10
UPnP SecurityConsole:1 Service	ISO/IEC 29341-13-11

1. Overview and Scope

This service template is compliant with the UPnP Device Architecture version 1.0. It defines a service type referred to herein as RenderingControl*:1*.

1.1. Introduction

Most rendering devices contain a number of dynamically configurable attributes that affect how the current content is rendered. For example, video rendering devices, such as TVs, allow user control of display characteristics such as brightness and contrast, whereas audio rendering devices allow control of audio characteristics such as volume, balance, equalizer settings, etc. The RenderingControl*:1* service is intended to provide Control Points with the ability to query and/or adjust any rendering attribute that the device supports.

The RenderingControl*:1* service enables a Control Point to:

- Discover the set of attributes supported by the device.
- Retrieve the current setting of any supported attribute
- Change the setting of (e.g. control) any modifiable attribute
- Restore the settings defined by a named Preset

The RenderingControl*:1* service DOES NOT:

- Control the flow of the associated content (e.g. Play, Stop, Pause, Seek, etc.).
- Provide a mechanism to enumerate locally stored content.
- Provide a mechanism to select the content that is to be rendered.
- Provide a mechanism to send content to another device (via the home network or direct connection).

1.2. Multi-input Devices

Some high-end AV device are capable of receiving multiple pieces of content at the same time and combining that content together so that it can be rendered together using a single set of output hardware. For example, while displaying a TV program, high-end TVs can also display additional content (e.g. VCR content) in a PIP (Picture-In-Picture) window. Similarly, a Karaoke machine can mix together the background music with a singer's voice so that both sounds are played together on the same set of speakers.

As with all devices, the Rendering Control Service allows a Control Point to adjust the output characteristics of the post-mixed content before it is actually rendered. However, in many cases, Control Points may need to control the output characteristics of the individual input content before it is mixed together with the other input content. In order to support this, the Rendering Control Service includes an InstanceID parameter with each action that allows the Control Point to identify to which content the action is to be applied (e.g. the post-mixed content or one of the pre-mixed input content items).

By convention, an InstanceID of 0 indicates that the invoked action should be applied to the post-mixed content. Similarly, each pre-mixed input content is assigned a unique InstanceID whose value is a non-zero, positive integer. Refer to Section 2.5 (Theory of Operation) for additional information.

2. Service Modeling Definitions

2.1. Service Type

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

urn:schemas-upnp-org:service:RenderingControl:*1*

The shorthand RenderingControl:*1* is used herein to refer to this service type.

2.2. State Variables

Table 1: State Variables

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value[2]	Default Value[2]	Eng. Units
LastChange	*R*	*string*			
PresetNameList	*R*	*string*	CSV[3] (string)		
Brightness	*O*	*ui2*	*Min = 0* *Max = Vendor defined* *Step = 1*		
Contrast	*O*	*ui2*	*Min = 0* *Max = Vendor defined* *Step = 1*		
Sharpness	*O*	*ui2*	*Min = 0* *Max = Vendor defined* *Step = 1*		
RedVideoGain	*O*	*ui2*	*Min = 0* *Max = Vendor defined* *Step = 1*		
GreenVideoGain	*O*	*ui2*	*Min = 0* *Max = Vendor defined* *Step = 1*		
BlueVideoGain	*O*	*ui2*	*Min = 0* *Max = Vendor defined* *Step = 1*		
RedVideoBlackLevel	*O*	*ui2*	*Min = 0* *Max = Vendor defined* *Step = 1*		
GreenVideoBlackLevel	*O*	*ui2*	*Min = 0* *Max = Vendor defined* *Step = 1*		
BlueVideoBlackLevel	*O*	*ui2*	*Min = 0* *Max = Vendor defined* *Step = 1*		
ColorTemperature	*O*	*ui2*	*Min = 0* *Max = Vendor defined* *Step = 1*		
HorizontalKeystone	*O*	*i2*	*Min = Vendor defined (Must be <= 0)* *Max = Vendor defined* *Step = 1*		
VerticalKeystone	*O*	*i2*	*Min = Vendor defined (Must be <= 0)* *Max = Vendor defined* *Step = 1*		
Mute	*O*	*boolean*			

Volume	*O*	*ui2*	*Min = 0* *Max = Vendor defined* *Step = 1*		
VolumeDB ***Error! Bookmark not defined.***	*O*	*i2*	*Min = Vendor defined* *Max = Vendor defined*		*1/256 dB*
Loudness	*O*	*boolean*			
A_ARG_TYPE_Channel	*R*	*string*	See Table 2-15	*N/A*	*N/A*
A_ARG_TYPE_InstanceID	*R*	*ui4*		*N/A*	*N/A*
A_ARG_TYPE_PresetName	*R*	*string*		*N/A*	*N/A*
Non-standard state variables implemented by an UPnP vendor go here.	*X*	*TBD*	*TBD*	*TBD*	*TBD*

[1] R = Required, O = Optional, X = Non-standard.

[2] Values listed in this column are required. To specify standard optional values or to delegate assignment of values to the vendor, you must reference a specific instance of an appropriate table below.

[3] CSV stands for Comma-Separated Value list. The type between brackets denotes the UPnP data type used for the elements inside the list. CVS is defined more formally in the ContentDirectory service template.

[4] The Volume and VolumeDB state variables are defined as a pair. Therefore, each implementation of this service MUST either support both of them or support none of them. At all times, these two state variables must remain synchronized with each other (i.e. both of them must always represent the same volume setting).

Table 2-1: allowedValueRange for *Brightness*

	Value	Req. or Opt.
minimum	*0*	*R*
maximum	*Vendor defined*	*R*
step	*1*	*R*

Table 2-2: allowedValueRange for *Contrast*

	Value	Req. or Opt.
minimum	*0*	*R*
maximum	*Vendor defined*	*R*
step	*1*	*R*

Table 2-3: allowedValueRange for *Sharpness*

	Value	Req. or Opt.
minimum	*0*	*R*
maximum	*Vendor defined*	*R*
step	*1*	*R*

Table 2-4: allowedValueRange for *RedVideoGain*

	Value	Req. or Opt.
minimum	*0*	*R*
maximum	*Vendor defined*	*R*
step	*1*	*R*

Table 2-5: allowedValueRange for *GreenVideoGain*

	Value	Req. or Opt.
minimum	*0*	*R*
maximum	*Vendor defined*	*R*
step	*1*	*R*

Table 2-6: allowedValueRange for *BlueVideoGain*

	Value	Req. or Opt.
minimum	*0*	*R*
maximum	*Vendor defined*	*R*
step	*1*	*R*

Table 2-7: allowedValueRange for *RedVideoBlackLevel*

	Value	Req. or Opt.
minimum	*0*	*R*
maximum	*Vendor defined*	*R*
step	*1*	*R*

Table 2-8: allowedValueRange for *GreenVideoBlackLevel*

	Value	Req. or Opt.
minimum	*0*	*R*
maximum	*Vendor defined*	*R*
step	*1*	*R*

Table 2-9: allowedValueRange for *BlueVideoBlackLevel*

	Value	Req. or Opt.
minimum	*0*	*R*
maximum	*Vendor defined*	*R*
step	*1*	*R*

Table 2-10: allowedValueRange for *ColorTemperature*

	Value	Req. or Opt.
minimum	*0*	*R*
maximum	*Vendor defined*	*R*
step	*1*	*R*

Table 2-11: allowedValueRange for *HorizontalKeystone*

	Value	Req. or Opt.
minimum	*Vendor defined (Must be <= 0)*	*R*
maximum	*Vendor defined*	*R*
Step	*1*	*R*

Table 2-12: allowedValueRange for *VerticalKeystone*

	Value	Req. or Opt.
minimum	*Vendor defined (Must be <= 0)*	*R*
maximum	*Vendor defined*	*R*
step	*1*	*R*

Table 2-13: allowedValueRange for *Volume*

	Value	Req. or Opt.
minimum	0	*R*
maximum	*Vendor defined*	*R*
Step	*1*	*R*

Table 2-14: allowedValueRange for *VolumeDB*

	Value	Req. or Opt.
minimum	*Vendor defined*	*R*
maximum	*Vendor defined*	*R*
step	*Vendor defined*	*O*

Table 2-15*:* allowedValueList for *A_ARG_TYPE_Channel*

Value	Req. or Opt.
Master	*R*
LF	*O*
RF	*O*
CF	*O*
LFE	*O*
LS	*O*
RS	*O*
LFC	*O*
RFC	*O*
SD	*O*
SL	*O*
SR	*O*
T	*O*
B	*O*
Vendor-defined	*O*

Table 2-16: allowedValueList for *A_ARG_TYPE_PresetName*

Value	Req. or Opt.
"FactoryDefaults"	*R*
"InstallationDefaults"	*O*
Vendor defined	*O*

2.2.1. *LastChange*

This variable is used exclusively for eventing purposes to allow clients to receive meaningful event notifications whenever the state of the device changes. The LastChange state variable identifies all of the state variables that have changed since the last time the LastChange state variable was evented. Refer to Section 2.3.1 (Event Model) for additional information

The format of this state variable conforms to the XML schema described in Section 5 (“LastChange” State Variable Schema). The following XML document illustrates a typical example of the schema:

```
<Event xmlns=”urn:schemas-upnp-org:metadata-1-0/AVT_RCS">
        <InstanceID val=”0”>
                <Brightness val=”36”/>
                <Contrast val=”54”/>
                ...
         </InstanceID>
        <InstanceID val=”1”>
                <Mute channel=”Master”  val=”0”/>
                <Volume channel=”CF”val=”24”/>
                ...
         </InstanceID>
        ...
</Event>
```

As illustrated above, the LastChange state variable contains a single root element whose body contains one or more InstanceID elements, each corresponding to a (virtual) instance of the Rendering Control Service, whose state has changed since the last time the LastChange state variable was evented. Each InstanceID element contains one or more state variable elements that identify all of the state variables within that instance that changed. Each state variable element contains the new (current) value of that state variable.

In the example above, the Brightness and Contrast setting of instance 0 has changed to 36 and 54, respecitively. Additionally, the Mute setting on the Master channel of instance 1 has been set to 0 (false) (i.e. muting has been turned off) and the Volume of the Center Front channel of instance 1 has been set to 24. Note: That only the audio-related state variable include a ‘channel’ attribute, which identifies the audio channel that has experienced a change.

When a given state variable (within the same instance) changes multiple times before the moderation period of the LastChange state variable expires, only one state variable element will appear within the InstanceID element. The previous state variable element must be removed and replaced with a new state variable element that reflects the most recent value of that state variable.

State variable elements may appear in any order within a given InstanceID element. This implies that no meaning may be deduced from the order in which the state variables for a given instance are listed. Similarly, the order of InstanceID elements has no particular meaning and may appear in any order.

For example, when the Brightness of instance 0 changes from 26 to 54 then to 48, and the Brightness of instance 1 changes from 54 to 35, then to 11, LastChange is set to the following:

```
<Event xmlns=”urn:schemas-upnp-org:metadata-1-0/AVT_RCS">
        <InstanceID val=”0”><Brightness val=”48”/></InstanceID>
        <InstanceID val=”1”><Brightness val=”11”/></InstanceID>
</Event>
```

The LastChange state variable is the only state variable that is evented using the standard UPnP event mechanism.. All other state varaibes are indirectly evented via the LastChange state variable event. Refer to Section 2.3.1 (Event Model) for additional details.

2.2.2. *PresetNameList*

This variable contains a comma-separated list (i.e. CSV (string) see table footnote 3 in Table 1) of valid preset names currently supported by this device. Its value changes if/when the device changes the set of presets that it supports. This may occur in conjunction with a vendor-defined action or some other non-UPnP event. This state variable will include any of the predefined presets that are supported by the device.

2.2.3. *Brightness*

This unsigned integer variable represents the current brightness setting of the associated display device. Its value ranges from a minimum of 0 to some device specific maximum. A numerical change of 1 corresponds to the smallest incremental change that is supported by the device.

Higher values increase the brightness of the display's output.

2.2.4. *Contrast*

This unsigned integer variable represents the current contrast setting of the associated display device. Its value ranges from a minimum of 0 to some device specific maximum. A numerical change of 1 corresponds to the smallest incremental change that is supported by the device.

Higher values increase the contrast of the display's output.

2.2.5. *Sharpness*

This unsigned integer variable represents the current sharpness setting of the associated display device. Its value ranges from a minimum of 0 to some device specific maximum. A numerical change of 1 corresponds to the smallest incremental change that is supported by the device.

Higher values accentuate fine detail.

2.2.6. *RedVideoGain*

This unsigned integer variable represents the current setting of the red "gain" control for the associated display device. Its value ranges from a minimum of 0 to some device specific maximum. A numerical change of 1 corresponds to the smallest incremental change that is supported by the device.

Higher values increase the intensity of red in the display's output. See Section 2.2.22 for additional information.

2.2.7. *GreenVideoGain*

This unsigned integer variable represents the current setting of the green "gain" control for the associated display device. Its value ranges from a minimum of 0 to some device specific maximum. A numerical change of 1 corresponds to the smallest incremental change that is supported by the device.

Higher values increase the intensity of green in the display's output. See Section 2.2.22 for additional information.

2.2.8. *BlueVideoGain*

This unsigned integer variable represents the current setting of the blue "gain" control for the associated display device. Its value ranges from a minimum of 0 to some device specific maximum. A numerical change of 1 corresponds to the smallest incremental change that is supported by the device.

Higher values increase the intensity of blue in the display's output. See Section 2.2.22 for additional information.

2.2.9. *RedVideoBlackLevel*

This unsigned integer variable represents the current setting for the minimum output intensity of red for the associated display device. Its value ranges from a minimum of 0 to some device specific maximum. A numerical change of 1 corresponds to the smallest incremental change that is supported by the device.

Higher values increase the minimum output intensity of red in the display's output. See Section 2.2.22 for additional information.

2.2.10. *GreenVideoBlackLevel*

This unsigned integer variable represents the current setting for the minimum output intensity of green for the associated display device. Its value ranges from a minimum of 0 to some device specific maximum. A numerical change of 1 corresponds to the smallest incremental change that is supported by the device.

Higher values increase the minimum output intensity of green in the display's output. See Section 2.2.22 for additional information.

2.2.11. *BlueVideoBlackLevel*

This unsigned integer variable represents the current setting for the minimum output intensity of blue for the associated display device. Its value ranges from a minimum of 0 to some device specific maximum. A numerical change of 1 corresponds to the smallest incremental change that is supported by the device.

Higher values increase the minimum output intensity of blue in the display's output. See Section 2.2.22 for additional information.

2.2.12. *ColorTemperature*

This unsigned integer variable represents the current setting for the "color quality" of white for the associated display device. Its value ranges from a minimum of 0 to some device specific maximum. A numerical change of 1 corresponds to the smallest incremental change that is supported by the device

Lower values produce "warmer" shades of white (e.g. biased towards yellow/orange) and higher values produce "cooler" shades of white (e.g. biased towards blue).

2.2.13. *HorizontalKeystone*

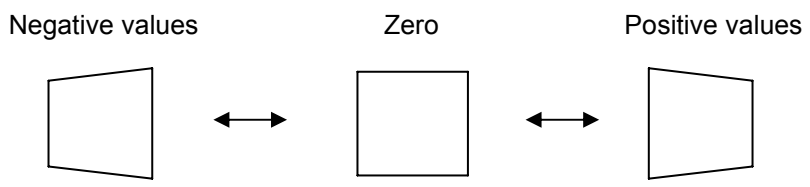
This signed integer variable represents the current level of compensation for horizontal distortion (described below) of the associated display device. Its value ranges from device-specific negative number to a device specific positive number. Zero does not need to be in the middle of this range, although it will be for most devices. A numerical change of 1 corresponds to the smallest incremental change that is supported by the device.

Horizontal distortion can occur when the display device is horizontally misaligned from the center of the viewing screen. For example, when a video/still-image projection device is shifted to the left or right of the display screen, the image becomes distorted (one side is taller than the other). The Horizontal Keystone state variable is used to compensate for this type of distortion.

Note: The following descriptions illustrate the effect of the HorizontalKeystone state variable when applied to a projection device that is properly centered with respect to the viewing screen (e.g. no misalignment distortion exists).

The left side of the display's output decreases in size as the value becomes more negative (i.e. moves farther away from zero). This produces a trapezoidal-shape image with the left and right edges remaining parallel, but with the left side shorter than the right side.

The right side of the display's output decreases in size as the value increases (i.e. moves farther from zero in a positive direction). This produces a trapezoidal-shape image with the left and right edge remaining parallel, but with the right side shorter than the left side.



2.2.14. *VerticalKeystone*

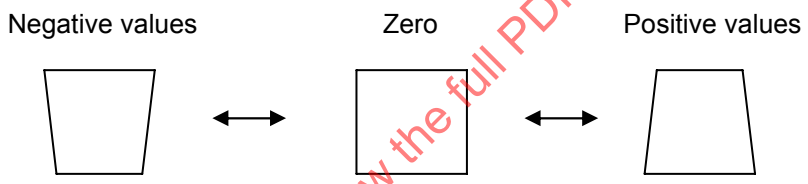
This signed integer variable represents the current level of compensation for vertical distortion (described below) of the associated display device. Its value ranges from device-specific negative number to a device specific positive number. Zero does not need to be in the middle of this range, although it will be for most devices. A numerical change of 1 corresponds to the smallest incremental change that is supported by the device.

Vertical distortion can occur when the display device is vertical misaligned from the center of the viewing screen. For example, when a video/still-image projection device is moved above or below the display screen, the image becomes distorted (i.e. the top and bottom edges have different lengths). The Vertical Keystone state variable is used to compensate for this type of distortion.

Note: The following descriptions illustrate the effect of the VerticalKeystone state variable when applied to a projection device that is properly centered with respect to the viewing screen (e.g. no misalignment distortion exists).

The bottom edge of the display's output decreases in size as the value becomes more negative (i.e. moves farther away from zero). This produces a trapezoidal-shape image with the top and bottom edges remaining parallel, but with the bottom edge shorter than the top edge.

The top edge of the display's output decreases in size as the value increases (i.e. moves farther from zero in a positive direction). This produces a trapezoidal-shape image with the top and bottom edges remaining parallel, but with the top edge shorter than the bottom edge.



2.2.15. *Mute*

This boolean variable represents the current "mute" setting of the associated audio channel. A value of TRUE (e.g. a numerical value of 1) indicates that the output of the associated audio channel is currently muted (i.e. that channel is not producing any sound).

2.2.16. *Volume*

This unsigned integer variable represents the current volume setting of the associated audio channel. Its value ranges from a minimum of 0 to some device specific maximum. A numerical change of 1 corresponds to the smallest incremental change that is supported by the device

Lower values produce a quieter sound and higher values produce a louder sound. A value of 0 represents complete silence (on the associated channel) and a value of 1 represents the quietest level of sound supported by the device for that channel.

2.2.17. *VolumeDB*

This signed integer variable represents the current volume setting of the associated audio channel. Its value represents the current setting in units of "1/256 of a decibel (dB)". This means that a numeric change of 1 corresponds to a volume change of 1/256 dB. The value range for this variable is a minimum value of -32,767 (8001_{16}) (which equals–127.9961dB) and a maximum value of +32,767 ($7FFF_{16}$) (which equals +127.9961dB). The value corresponding to 8000_{16} is invalid.

Each implementation of this service must specify a vendor-defined minimum and maximum value. Most implementations will not support all incremental values between the supported minimum and maximum values. If a Control Point attempts to set VolumeDB to an unsupported value (within the specified minimum and

maximum values), the device will set the volume to the closest setting that is supported by the device for the associated channel.

Lower values (including negative values) produce a quieter sound and larger values produce a louder sound. Setting the volume to the minimum value supported by the device produces complete silence on that channel.

2.2.18. *Loudness*

This boolean variable represents the current "loudness" setting of the associated audio channel. A value of TRUE (e.g. a numerical value of 1) indicates that the loudness effect is active.

2.2.19. *A_ARG_TYPE_Channel*

This variable is used to identify a particular channel of an audio output stream. A channel, except the Master channel, is associated with the location of the speaker where the audio data stream is to be presented. It is customary to refer a channel using the spatial position of associated speaker as described below.

The Master channel is a logical channel and, therefore, has no spatial position associated with it. A one-channel channel cluster does not have spatial position associated with it either and will use the Master channel to control its properties.

The following channel spatial positions are defined:

- Master (Master)
- Left Front (LF)
- Right Front (RF)
- Center Front (CF)
- Low Frequency Enhancement (LFE) [Super woofer]
- Left Surround (LS)
- Right Surround (RS)
- Left of Center (LFC) [in front]
- Right of Center (RFC) [in front]
- Surround (SD) [rear]
- Side Left (SL) [left wall]
- Side Right (SR) [right wall]
- Top (T) [overhead]
- Bottom (B) [bottom]

A channel cluster is the collection of all channels, including the Master channel, within an audio stream. A single channel (mono) cluster has only one channel – the Master channel. A two-channel (stereo) cluster has three channels – the Master channel, the Left Front channel, and the Right Front channel. In this specification, only the Master channel is required. All other channels are optional, see Table 2-15 for details.

2.2.20. *A_ARG_TYPE_InstanceID*

This signed integer parameter is used to specify the logical instance of the Rendering Control Service to which the associated action should be applied. A value of '0' indicates that the action should be applied to the global (post-mix) stream, as shown in Figure 1.

2.2.21. *A_ARG_TYPE_PresetName*

This string parameter is used to specify the name of a device preset. This may include any of the names listed in the PresetNameList state variable or any of the predefined presets (listed below) that are supported by the device.

Table 2-17: Predefined names of some common presets

Value	Definition
FactoryDefaults	The factory settings defined by the device's manufacturer.
InstallationDefaults	The installation settings defined by the installer of the

	device. This may or may not be the same as the Factory defaults.

2.2.22. Relationships Between State Variables

Except for the LastChange and two volume-related state variables (e.g. Volume and VolumeDB), all state variables operate independently. However, whenever any (non A_ARG_TYPE_xxx) state variable changes, a state change descriptor is added to the LastChange state variable to reflect the modified state. Refer to the description of the LastChange state variable and overall eventing model for more details.

xxxVideoGain and xxxVideoBlackLevel:

As described in Section 2.2.6 - 2.2.11 (*RedVideoGain* thru *BlueVideoBlackLevel*), a pair of state variables is defined to control the output intensity of each primary color (i.e. red, green, and blue). The state variable pair associated with each color includes one state variable to control the "gain" of that color and one state variable to control the minimum output intensity of that color (e.g. *RedVideoGain* and *RedVideoBlackLevel*).

Although these two state variables are associated with the same color, they function independently from each other (e.g. changing the value of one state variable does not affect the value of the other). However, each state variable within a given pair (i.e. associated with a given color) may affect the output intensity of that color. For example, while displaying a static image, increasing the value of either the *RedVideoGain* or *RedVideoBlackLevel* state variables may cause an increase the amount of red that is displayed. Similarly, decreasing either state variable may cause a decrease in the amount of red.

The precise effect of these two variables may vary from device to device. However, as a common example, implementation the *RedVideoGain* controls the multiplication factor between the input and output intensity of the color red, and the *RedVideoBlackLevel* controls minimum output intensity of red regardless of the input intensity. Thus, a typical implementation may be described using a variation of the following formula:

Red Output Intensity = *RedVideoGain* * Red Input Intensity + *RedVideoBlackLevel*

Volume and VolumeDB

The Volume and VolumeDB state variables are directly related to one another. Both state variables reflect the current volume setting, but each one does so using a different numeric scale. The Volume state variable uses a linear numeric scale in which each discreet volume setting that is supported by the device is represented by a contiguous range of values from 0 to some maximum value. This maximum value indicates the total number of discrete settings that the device supports. In contrast, the VolumeDB state variable uses a linear numeric scale based on 1/256dB engineering units. This provides a mechanism to achieve a uniform volume setting regardless between different devices. Note: The same VolumeDB setting on two different devices will not necessarily produce the same loudness of sound.

2.3. Eventing and Moderation

As the table below summarizes, the RenderingControl:*1* specification uses moderated eventing for only one of its standard state variables and no eventing for the rest.

Table 2: Event moderation

Variable Name	Evented	Moderated Event	Max Event Rate [1]	Logical Combination	Min Delta per Event [2]
LastChange	*Yes*	*Yes*	*0.2*		
PresetNameList	*No*				
Brightness	*No*				
Contrast	*No*				

Sharpness	*No*				
RedVideoGain	*No*				
GreenVideoGain	*No*				
BlueVideoGain	*No*				
RedVideoBlackLevel	*No*				
GreenVideoBlackLevel	*No*				
BlueVideoBlackLevel	*No*				
ColorTemperature	*No*				
HorizontalKeystone	*No*				
VerticalKeystone	*No*				
Mute	*No*				
Volume	*No*				
VolumeDB	*No*				
Loudness	*No*				
A_ARG_TYPE_Channel	*No*				
A_ARG_TYPE_InstanceID	*No*				
A_ARG_TYPE_PresetName	*No*				
Non-standard state variables implemented by an UPnP vendor go here.	*TBD*	*TBD*	*TBD*	*TBD*	*TBD*

[1] Determined by N, where Rate = (Event)/(N secs).
[2] (N) * (allowedValueRange Step).

2.3.1. Event Model

Since the Rendering Control Service supports multiple logical instances (via the InstanceID parameter included in each action), the traditional UPnP eventing model is unable to differentiate between multiple instances of the same state variable. Therefore, the Rendering Control Service event model defines a specialized state variable (LastChange) that is used exclusively for eventing individual state changes. In this model,the LastChange state change is the only variable that is evented using the standard UPnP event mechanism. All other state variables are indirectly evented via the LastChange state variable. (Note: A_ARG_TYPE_ state variables are not evented, either directly or indirectly.)

When the value of a state variable changes, information about that change is added to the LastChange state variable as described in Section 2.2.1 (*LastChange*). As a result of modifying the LastChange state variable, its new value (i.e. the information describing the original state change) is evented using the standard UPnP eventing mechanism. In this manner, the change to the original state variable is indirectly evented to all interested Control Points.

Since the LastChange state variable is a moderated state variable, multiple state changes are accumulated in the LastChange state variable until its moderation period expires. When this occurs, the current value of the LastChange state variable is sent out using the standard UPnP eventing mechanism. This notification informs interested Control Points of all state changes that have occurred since the previous LastChange event was sent.

After the LastChange state variable is evented, its contents are cleared out in preparation for the next state change. (Note: The act of clearing out the “stale” contents of the LastChange state variable does not need to generate another event notification even though its value has changed. Doing so would only generate unnecessary network traffic.). Because the LastChange state variable is moderated, a given state variable may change multiple times before the current moderation period expires. In such cases, the LastChange state variable will contain a single entry for that state variable reflecting its current (most recent) value.

The standard UPnP event mechanism indicates that when a Control Point subscribes to receive events, the current values of all evented state variables are returned to the subscriber. However, since the LastChange state

variable is the only state variable that is directly evented (i.e. all other state variables are indirectly evented via the LastChange state variable), it isn't very meaningful to respond to an event subscription request with the current value of the LastChange state variable. Its value is too transitory to be of any use to the subscribing Control Point. Therefore, when an event subscription is received, the device should respond with the current values of all (indirectly evented) state variables within all valid instances of this service. Refer to Section 2.2.1 (*LastChange*) for additional information.

2.4. Actions

The following tables and subsections define the various RenderingControl:*1* actions. Except where noted, if an invoked action returns an error, the state of the device will be unaffected.

Table 3: Actions

Name	Req. or Opt. [1]
ListPresets	*R*
SelectPreset	*R*
GetBrightness	*O*
SetBrightness	*O*
GetContrast	*O*
SetContrast	*O*
GetSharpness	*O*
SetSharpness	*O*
GetRedVideoGain	*O*
SetRedVideoGain	*O*
GetGreenVideoGain	*O*
SetGreenVideoGain	*O*
GetBlueVideoGain	*O*
SetBlueVideoGain	*O*
GetRedVideoBlackLevel	*O*
SetRedVideoBlackLevel	*O*
GetGreenVideoBlackLevel	*O*
SetGreenVideoBlackLevel	*O*
GetBlueVideoBlackLevel	*O*
SetBlueVideoBlackLevel	*O*
GetColorTemperature	*O*
SetColorTemperature	*O*
GetHorizontalKeystone	*O*
SetHorizontalKeystone	*O*
GetVerticalKeystone	*O*
SetVerticalKeystone	*O*
GetMute	*O*
SetMute	*O*
GetVolume	*O*
SetVolume	*O*
GetVolumeDB	*O*

SetVolumeDB	*O*
GetVolumeDBRange	*O*
GetLoudness	*O*
SetLoudness	*O*
Non-standard actions implemented by an UPnP vendor go here.	*X*

[1] R = Required, O = Optional, X = Non-standard.

2.4.1. *ListPresets*

This action returns a list of the currently defined presets. The CurrentPresetNameList "out" parameter contains a comma-separated list (i.e. CSV [2] (string)) of preset names that include both predefined (static) presets and user-defined (dynamically created) presets that may be created via a private vendor-defined action.

2.4.1.1. Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
CurrentPresetNameList	*OUT*	*PresetNameList*

2.4.1.2. Effect on State

This action does not affect the state of the service.

2.4.1.3. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.2. *SelectPreset*

This action restores (a subset) of the state variables to the values associated with the specified preset. The specified preset name may be one of the predefined presets (listed in Table 2-17) or one of the user-defined presets that may have been created via a private vendor-defined action. The selected preset determines which state variables will be affected.

2.4.2.1. Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
PresetName	*IN*	*A_ARG_TYPE_PresetName*

2.4.2.2. Effect on State

This action sets the state of the service to the values associated with the specified preset.

[2] CSV stands for Comma-Separated Value list. The type between brackets denotes the UPnP data type used for the elements inside the list. CVS is defined more formally in the ContentDirectory service template.

2.4.2.3. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
701	Invalid Name	The specified name is not a valid preset name.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.3. GetBrightness

This action retrieves the current value of the Brightness state variable of the specified instance of this service.

2.4.3.1. Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
CurrentBrightness	*OUT*	*Brightness*

2.4.3.2. Effect on State

This action does not affect the state of the service.

2.4.3.3. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.4. SetBrightness

This action sets the Brightness state variable of the specified instance of this service to the specified value.

2.4.4.1. Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	IN	A_ARG_TYPE_InstanceID
DesiredBrightness	IN	Brightness

2.4.4.2. Effect on State

This action affects the Brightness state variable of the specified instance of this service.

2.4.4.3. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.5. GetContrast

This action retrieves the current value of the Contrast state variable of the specified instance of this service.

2.4.5.1. Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
CurrentContrast	*OUT*	*Contrast*

2.4.5.2. Effect on State

This action does not affect the state of the service.

2.4.5.3. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.6. SetContrast

This action sets the Contrast state variable of the specified instance of this service to the specified value.

2.4.6.1. Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
DesiredContrast	*IN*	*Contrast*

2.4.6.2. Effect on State

This action affects the Contrast state variable of the specified instance of this service.

2.4.6.3. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.7. GetSharpness

This action retrieves the current value of the Sharpness state variable of the specified instance of this service.

2.4.7.1. Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
CurrentSharpness	*OUT*	*Sharpness*

2.4.7.2. Effect on State

This action does not affect the state of the service.

2.4.7.3. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.8. *SetSharpness*

This action sets the Sharpness state variable of the specified instance of this service to the specified value.

2.4.8.1. Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
DesiredSharpness	*IN*	*Sharpness*

2.4.8.2. Effect on State

This action affects the Sharpness state variable of the specified instance of this service.

2.4.8.3. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.9. *GetRedVideoGain*

This action retrieves the current value of the RedVideoGain state variable of the specified instance of this service.

2.4.9.1. Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
CurrentRedVideoGain	*OUT*	*RedVideoGain*

2.4.9.2. Effect on State

This action does not affect the state of the service.

2.4.9.3. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.10. SetRedVideoGain

This action sets the RedVideoGain state variable of the specified instance of this service to the specified value.

2.4.10.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
DesiredRedVideoGain	*IN*	*RedVideoGain*

2.4.10.2.Effect on State

This action affects the RedVideoGain state variable of the specified instance of this service.

2.4.10.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.11. GetGreenVideoGain

This action retrieves the current value of the GreenVideoGain state variable of the specified instance of this service.

2.4.11.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
CurrentGreenVideoGain	*OUT*	*GreenVideoGain*

2.4.11.2.Effect on State

This action does not affect the state of the service.

2.4.11.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.12. SetGreenVideoGain

This action sets the GreenVideoGain state variable of the specified instance of this service to the specified value.

2.4.12.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
DesiredGreenVideoGain	*IN*	*GreenVideoGain*

2.4.12.2.Effect on State

This action affects the GreenVideoGain state variable of the specified instance of this service.

2.4.12.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.13. *GetBlueVideoGain*

This action retrieves the current value of the BlueVideoGain state variable of the specified instance of this service.

2.4.13.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
CurrentBlueVideoGain	*OUT*	*BlueVideoGain*

2.4.13.2.Effect on State

This action does not affect the state of the service.

2.4.13.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.14. *SetBlueVideoGain*

This action sets the BlueVideoGain state variable of the specified instance of this service to the specified value.

2.4.14.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
DesiredBlueVideoGain	*IN*	*BlueVideoGain*

2.4.14.2.Effect on State

This action affects the BlueVideoGain state variable of the specified instance of this service.

2.4.14.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.15. *GetRedVideoBlackLevel*

This action retrieves the current value of the RedVideoBlackLevel state variable of the specified instance of this service.

2.4.15.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
CurrentRedVideoBlackLevel	*OUT*	*RedVideoBlackLevel*

2.4.15.2.Effect on State

This action does not affect the state of the service.

2.4.15.3.Errors

ErrorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.16. *SetRedVideoBlackLevel*

This action sets the RedVideoBlackLevel state variable of the specified instance of this service to the specified value.

2.4.16.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
DesiredRedVideoBlackLevel	*IN*	*RedVideoBlackLevel*

2.4.16.2.Effect on State

This action affects the RedVideoBlackLevel state variable of the specified instance of this service.

2.4.16.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.17. GetGreenVideoBlackLevel

This action retrieves the current value of the GreenVideoBlackLevel state variable of the specified instance of this service.

2.4.17.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
CurrentGreenVideoBlackLevel	*OUT*	*GreenVideoBlackLevel*

2.4.17.2.Effect on State

This action does not affect the state of the service.

2.4.17.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.18. SetGreenVideoBlackLevel

This action sets the GreenVideoBlackLevel state variable of the specified instance of this service to the specified value.

2.4.18.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
DesiredGreenVideoBlackLevel	*IN*	*GreenVideoBlackLevel*

2.4.18.2.Effect on State

This action affects the GreenVideoBlackLevel state variable.

2.4.18.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.19. GetBlueVideoBlackLevel

This action retrieves the current value of the BlueVideoBlackLevel state variable of the specified instance of this service.

2.4.19.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
CurrentBlueVideoBlackLevel	*OUT*	*BlueVideoBlackLevel*

2.4.19.2.Effect on State

This action does not affect the state of the service.

2.4.19.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.20. SetBlueVideoBlackLevel

This action sets the BlueVideoBlackLevel state variable of the specified instance of this service to the specified value.

2.4.20.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
DesiredBlueVideoBlackLevel	*IN*	*BlueVideoBlackLevel*

2.4.20.2.Effect on State

This action affects the BlueVideoBlackLevel state variable of the specified instance of this service.

2.4.20.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.21. GetColorTemperature

This action retrieves the current value of the ColorTemperature state variable of the specified instance of this service.

2.4.21.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
CurrentColorTemperature	*OUT*	*ColorTemperature*

2.4.21.2.Effect on State

This action does not affect the state of the service.

2.4.21.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.22. SetColorTemperature

This action sets the ColorTemperature state variable of the specified instance of this service to the specified value.

2.4.22.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
DesiredColorTemperature	*IN*	*ColorTemperature*

2.4.22.2.Effect on State

This action affects the ColorTemperature state variable of the specified instance of this service.

2.4.22.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.23. GetHorizontalKeystone

This action retrieves the current value of the HorizontalKeystone state variable of the specified instance of this service.

2.4.23.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
CurrentHorizontalKeystone	*OUT*	*HorizontalKeystone*

2.4.23.2.Effect on State

This action does not affect the state of the service.

2.4.23.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.24. SetHorizontalKeystone

This action sets the HorizontalKeystone state variable of the specified instance of this service to the specified value.

2.4.24.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
DesiredHorizontalKeystone	*IN*	*HorizontalKeystone*

2.4.24.2.Effect on State

This action affects the HorizontalKeystone state variable of the specified instance of this service.

2.4.24.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.25. GetVerticalKeystone

This action retrieves the current value of the VerticalKeystone state variable of the specified instance of this service.

2.4.25.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
CurrentVerticalKeystone	*OUT*	*VerticalKeystone*

2.4.25.2.Effect on State

This action does not affect the state of the service.

2.4.25.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.26. SetVerticalKeystone

This action sets the VerticalKeystone state variable of the specified instance of this service to the specified value.

2.4.26.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
DesiredVerticalKeystone	*IN*	*VerticalKeystone*

2.4.26.2.Effect on State

This action affects the VerticalKeystone state variable of the specified instance of this service.

2.4.26.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.27. GetMute

This action retrieves the current value of the Mute setting of the channel for the specified instance of this service.

2.4.27.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
Channel	*IN*	*A_ARG_TYPE_Channel*
CurrentMute	*OUT*	*Mute*

2.4.27.2.Effect on State

This action does not affect the state of the service.

2.4.27.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.28. SetMute

This action sets the Mute state variable of the specified instance of this service to the specified value.

2.4.28.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
Channel	*IN*	*A_ARG_TYPE_Channel*
DesiredMute	*IN*	*Mute*

2.4.28.2.Effect on State

This action affects the Mute state variable of the specified instance of this service.

2.4.28.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.29. GetVolume

This action retrieves the current value of the Volume state variable of the specified channel for the specified instance of this service. The CurrentVolume (OUT) parameter contains a value ranging from 0 to a device-specific maximum. See Section 2.2.16 (*Volume*) for more details,

2.4.29.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
Channel	*IN*	*A_ARG_TYPE_Channel*
CurrentVolume	*OUT*	*Volume*

2.4.29.2.Effect on State

This action does not affect the state of the service.

2.4.29.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.30. SetVolume

This action sets the Volume state variable of the specified Instance and Channel to the specified value. The DesiredVolume input parameter contains a value ranging from 0 to a device-specific maximum. See Section 2.2.16 (*Volume*) for more details,

2.4.30.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
Channel	*IN*	*A_ARG_TYPE_Channel*
DesiredVolume	*IN*	*Volume*

2.4.30.2.Effect on State

This action affects the Volume and VolumeDB state variables of the specified instance and channel. Both state variables need to change consistently.

2.4.30.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.31. GetVolumeDB

This action retrieves the current value of the VolumeDB state variable of the channel for the specified instance of this service. The CurrentVolume (OUT) parameter represents the current volume setting in units of 1/256 decibels (dB). See Section 2.2.17 (*VolumeDB*) for more details,

2.4.31.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
Channel	*IN*	*A_ARG_TYPE_Channel*
CurrentVolume	*OUT*	*VolumeDB*

2.4.31.2.Effect on State

This action does not affect the state of the service.

2.4.31.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.32. SetVolumeDB

This action sets the VolumeDB state variable of the specified Instance and Channel to the specified value. The DesiredVolume parameter represents the desired volume setting in unit of 1/256 decibels (dB). See Section 2.2.17 (*VolumeDB*) for more details.

2.4.32.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
Channel	*IN*	*A_ARG_TYPE_Channel*
DesiredVolume	*IN*	*VolumeDB*

2.4.32.2.Effect on State

This action affects the Volume and VolumeDB state variables of the specified instance and channel. Both state variables need to change consistently.

2.4.32.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.33. GetVolumeDBRange

This action retrieves the valid range for the VolumeDB state variable of the channel for the specified instance of this service. The MinValue and MaxValue (OUT) parameter identify the range of valid values for the VolumeDB state variable in units of 1/256 decibels (dB). See Section 2.2.17 (*VolumeDB*) for more details.

2.4.33.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
Channel	*IN*	*A_ARG_TYPE_Channel*
MinValue	*OUT*	*VolumeDB*
MaxValue	*OUT*	*VolumeDB*

2.4.33.2.Effect on State

This action does not affect the state of the service.

2.4.33.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.34. GetLoudness

This action retrieves the current value of the Loudness setting of the channel for the specified instance of this service.

2.4.34.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
Channel	*IN*	*A_ARG_TYPE_Channel*
CurrentLoudness	*OUT*	*Loudness*

2.4.34.2.Effect on State

This action does not affect the state of the service.

2.4.34.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.35. *SetLoudness*

This action sets the specified value of the Loudness state variable of the channel for the specified instance of this service to the specified value.

2.4.35.1.Arguments

Argument(s)	Direction	relatedStateVariable
InstanceID	*IN*	*A_ARG_TYPE_InstanceID*
Channel	*IN*	*A_ARG_TYPE_Channel*
DesiredLoudness	*IN*	*Loudness*

2.4.35.2.Effect on State

This action affects the Loudness state variable of the specified instance of this service.

2.4.35.3.Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	Invalid InstanceID	The specified instanceID is invalid.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.4.36. Relationships Between Actions

There is no inherent relationship between any of the various actions. All actions may be called in any order.

2.4.37. Common Error Codes

The following table lists error codes common to actions for this service type. If an action results in multiple errors, the most-specific error should be returned.

errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture section on Control.
402	Invalid Args	See UPnP Device Architecture section on Control.
404	Invalid Var	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
600-699	TBD	Common action errors. Defined by UPnP Forum Technical Committee.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.5. Theory of Operation

2.5.1. Multi-input Devices

Many traditional rendering devices are capable of receiving and rendering only a single item of content at a time. For example, traditional TVs can only receive and display a single TV show at a time, and a stereo system can only play a single song at a time. However, more and more devices are able to receive and render multiple items of content at the same time even though there is only a single piece of rendering hardware (e.g. a single TV screen or a single set of speakers). This capability is known as as "mixing".

As an example, while watching TV, a small Picture-in-a-Picture (PIP) can be overlaid on top of the main TV show so that another TV show (or VCR tape) can be watched at the same time. Although the TV contains a single set of output hardware (e.g. a single screen), the TV can take multiple items of content, mix them together, and render them both on the one screen.. Similarly, a karaoke system takes a singer's voice, mixes it with some background music, and renders it on a single set of speakers.

In the examples above, these 'multi-content' devices support a fixed number of input streams. However, there are some devices that support an arbitrary number of input streams. For example, a PC can take some video content and 'mix' it with the PC's display and render it in its own PC window. Depending on the processing power of the PC, it can 'mix' together and render a variable number of video-windows.

In these examples, some advanced devices have the ability to control the rendering characteristics of each input item independently from each other as well as the 'post-mixed' stream. For example, with a karaoke device, the volume of the singer's voice and the volume of the background music can be adjusted independently. Additionally, the volume of the post-mixed stream (i.e. singer + background) can be adjusted as a whole without affecting the relative settings of each individual input stream.

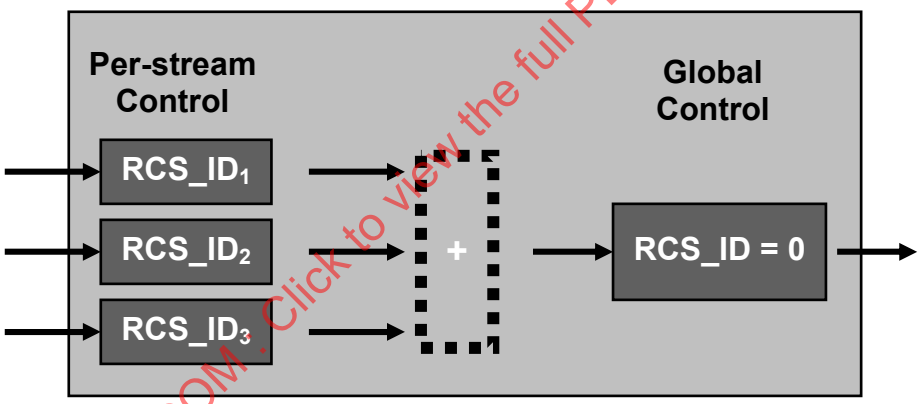


Figure 1: Some high-end, multiple input devices can 'mix' together multiple input streams and render the post mixed content on a single set of output hardware.

In order to support these types of devices, it is necessary for the Rendering Control Service (RCS) to support multiple (logical) instances of the service. As shown in Figure 1, a full-featured, high-end device that supports multiple input streams may assign a unique (logical) instance of the Rendering Control Service to each input stream. Each of these instances is assigned a unique ID (labeled RCS_ID in Figure 1) which can be used to adjust the rendering characteristics of its associated content without affecting any of the other input streams. Additionally, a "default" instance (ID=0) is assigned to control the rendering characteristics of the 'post-mixed' stream (i.e. all input content as a whole).

InstanceIDs are used by Control Points when invoking Rendering Control Service actions. The InstanceID parameter (included with each action) allows the Control Point to indicate to which 'stream' the invoked action should be applied. In order to control the rendering characteristics of an individual input stream (independently from all of the other streams) the Control Point uses the instance ID associated with that stream. In order to control the rendering characteristics of the 'post-mixed' stream, the Control Point uses InstanceID=0.

New (logical) instances of the Rendering Control Service (i.e. new InstanceIDs) are allocated outside of the Rendering Control Service using some external mechanism. As of this writing, only one allocation mechanism is defined. As described in Media Renderer device template, the device's ConnectionManager::PrepareForConnection() action assigns a InstanceID to the input stream (i.e. each connection) that is being prepared. The number of instances that a device can support depends on the device's implementation. (Refer to the Media Renderer device templates for additional information.).

As defined by the UPnP Device Architecture, a device's description document contains a single service description document for each (physical) service that is implemented by the device. However, when a device supports multiple (logical) instances of the Rendering Control Service, all of these (logical) instances are represented by the service description document for the one (physical) Rendering Control Service. In this case, the Rendering Control Service's description document reflects the actions and state variables supported by InstanceID=0. All other non-zero (logical) instances must support a subset of the actions and state variables supported by InstanceID=0. However, each non-zero instance may support a different subset of InstanceID=0 than the other non-zero (logical) instances. For those state variables that are supported by a non-zero instance, each instance must support the identical "allowedValueList" and/or "allowedValueRange" as InstanceID=0. If an unsupported action is invoked on a non-zero instance, the action will return error code 401 (Invalid Action).

As described in the Media Renderer device template, a rendering device that contains multiple, independent rendering hardware (e.g. two independent display screen, or two independent sets of output speakers) shall be modeled as multiple instantiations of the MediaRenderer device, each with its own Rendering Control, Connection Manager, and (optional) AV Transport services. In other words, from an UPnP AV modeling point of view, each output on a physical rendering device is treated as a completely independent Media Rendering device. Refer to the Media Renderer device template for more information.

2.5.2. Presets

Named presets allow a Control Point to put the device in a predetermined state in which certain state variables are set to predefined values. The set of currently available presets is listed in the PresetNameList state variable. Since a device is permitted to add or remove support for individual presets (e.g. in conjunction with a vendor–defined action), the PresetNameList state variable is (indirectly) evented as described in Section 2.3. Additionally, a Control Point can use the ListPresets() action to obtain an up to date list of supported presets.

A user can select one of the supported presets using the SelectPreset() action. This causes the device to set itself to a known state as defined associated by the selected preset. The exact definition of each preset is device-specific.

For informational purposes, Section 2.2.2 includes one possible definition for two vendor-specific actions to allow Control Points to dynamically create/destroy name presets. These actions are not part of this service template, but may be useful for those developers that wish to support this capability.

2.5.3. Controlling the Display of Visual Content

The Rendering Control Service exposes a number of state variables that allow Control Points to control the appearance of visual content. These include such characteristics as brightness, contrast, color intensity, etc. In order to control these characteristics, the Control Point simply invokes the appropriate action. For most of these actions, the desired setting of the display characteristics is passed in by the Control Point. In most cases, this parameter is a positive number between 0 and some device-specific maximum value. Each incremental value (i.e. an increase or decrease by one) corresponds to the smallest amount of change supported by the device.

Determine the current Brightness setting of the main display:

- Invoke GetBrightness() with an Instance ID of zero.
- A return value of 13 indicates that the display's Brightness is currently set to 13 "steps" (e.g. 13 device-specific increments) above the dimmest setting that is supported by the device.

Set the Brightness of the PIP display to the dimmest setting supported by the device :

- Invoke SetBrightness() with the PIP's InstanceID and a Brightness setting of 0.

2.5.4. Controlling Audio Content

The Rendering Control Service exposes a set of state variables that can be used to control the audio output of a device. These include various characteristics such as volume, mute, and loudness. However, unlike most visual

content, audio content is typically composed of one or more channels (e.g. a left and right channel). The Rendering Control service allows Control Points to control each of these channels independently or as a whole. In order to accomplish this, it is necessary for the Control Point to identify the channel that is to be controlled. This is accomplished via the 'Channel' parameter included in each action that is associated with the audio portion of an input stream. Each channel is uniquely named as described in Section 2.2.19. The "master" channel allows a Control Point to control the "composite" (post-mixed) audio content as a whole.

When controlling the volume of a particular channel, Control Points can choose between two different representations of the volume setting. One representation uses the Volume state variable and the other representation uses the VolumeDB state variable. As described in Sections 2.2.16 and 2.2.17, the Volume state variable represents volume as a contiguous set of "positions" numbered from 0 to some device-specific maximum, and the VolumeDB state variable represents volume in units of 1/256 of a decibel (dB). Two pair of actions (one pair for each representation) are provided to get and set the volume of a channel.

Set the volume of the audio content (as a whole) to the quietest (non-silent) setting:

- Invoke the SetVolume() action with the Channel parameter set to 'Master' and the DesiredVolume parameter set to 1.

Set the volume of the audio content (as a whole) 20 'notches/steps' higher than the current setting

- Invoke the GetVolume() action with the Channel parameter set to 'Master'. As a result of the previous example, the CurrentVolume out parameter returns a value of 1 indicating that the audio content is being rendered at volume position 1 i.e. the quietest non-silence setting supported by the device.

- Invoke the SetVolume() action with the Channel parameter set to "Master" and the DesiredVolume parameter set to 21 (1 + 20). This corresponds to the 20th quietest setting supported by the device.

Set the volume of the Center channel 5dB higher than the "Master" channel:

- Invoke the GetVolumeDB() action with the Channel parameter set to 'Master'. In this example, the CurrentVolume out parameter returns a value of -2560 (increments of 1/256dB), which indicates that the audio content is being rendered at -10dB. In this example, -10dB corresponds to volume position 21 as set in the previous example.

  Note: The correlation between the Volume and VolumeDB (i.e. 'position' settings and dB values) will vary from device to device. Additionally, for any given device, the relationship between Volume and VolumeDB may not be consistent throughout the range of valid values.

- Invoke the SetVolumeDB() action with the Channel parameter set to "CF" (for the Center Front channel) and the DesiredVolume parameter set to -1280 (-2560+1280 increments of 1/256dB), which corresponds to -5dB (-10+5).

Double the volume of the entire audio content:

(Note: Increasing the volume by 6dB doubles the volume level.)

- Invoke the GetVolumeDB() action with the Channel parameter set to 'Master'. Based on the previous example, the CurrentVolume out parameter returned a -1280 (increments of 1/256 dB), which indicates that the volume level is at -5dB.

- Invoke the SetVolumeDB() action with the Channel parameter set to "Master" (and the DesiredVolume parameter set to 256 (-1280+1536increments of 1/256dB), which corresponds to 1dB (-5+6).

3. XML Service Description

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action>
    <name>ListPresets</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>CurrentPresetNameList</name>
          <direction>out</direction>
          <relatedStateVariable>PresetNameList</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>SelectPreset</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>PresetName</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_PresetName</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetBrightness</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>CurrentBrightness</name>
          <direction>out</direction>
          <relatedStateVariable>Brightness</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>SetBrightness</name>
      <argumentList>
        <argument>
```

```
            <name>InstanceID</name>
            <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
          </argument>
          <argument>
            <name>DesiredBrightness</name>
            <direction>in</direction>
            <relatedStateVariable>Brightness</relatedStateVariable>
          </argument>
        </argumentList>
      </action>
      <action>
      <name>GetContrast</name>
        <argumentList>
          <argument>
            <name>InstanceID</name>
            <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
          </argument>
          <argument>
            <name>CurrentContrast</name>
            <direction>out</direction>
            <relatedStateVariable>Contrast</relatedStateVariable>
          </argument>
        </argumentList>
      </action>
      <action>
      <name>SetContrast</name>
        <argumentList>
          <argument>
            <name>InstanceID</name>
            <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
          </argument>
          <argument>
            <name>DesiredContrast</name>
            <direction>in</direction>
            <relatedStateVariable>Contrast</relatedStateVariable>
          </argument>
        </argumentList>
      </action>
      <action>
      <name>GetSharpness</name>
        <argumentList>
          <argument>
            <name>InstanceID</name>
            <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
          </argument>
          <argument>
            <name>CurrentSharpness</name>
            <direction>out</direction>
            <relatedStateVariable>Sharpness</relatedStateVariable>
          </argument>
        </argumentList>
      </action>
      <action>
      <name>SetSharpness</name>
        <argumentList>
```

```
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>DesiredSharpness</name>
          <direction>in</direction>
          <relatedStateVariable>Sharpness</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetRedVideoGain</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>CurrentRedVideoGain</name>
          <direction>out</direction>
          <relatedStateVariable>RedVideoGain</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>SetRedVideoGain</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>DesiredRedVideoGain</name>
          <direction>in</direction>
          <relatedStateVariable>RedVideoGain</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetGreenVideoGain</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>CurrentGreenVideoGain</name>
          <direction>out</direction>
          <relatedStateVariable>GreenVideoGain</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>SetGreenVideoGain</name>
```

```
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>DesiredGreenVideoGain</name>
          <direction>in</direction>
          <relatedStateVariable>GreenVideoGain</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetBlueVideoGain</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>CurrentBlueVideoGain</name>
          <direction>out</direction>
          <relatedStateVariable>BlueVideoGain</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>SetBlueVideoGain</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>DesiredBlueVideoGain</name>
          <direction>in</direction>
          <relatedStateVariable>BlueVideoGain</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
   <action>
    <name>GetRedVideoBlackLevel</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>CurrentRedVideoBlackLevel</name>
          <direction>out</direction>
          <relatedStateVariable>RedVideoBlackLevel</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
```

```
    <name>SetRedVideoBlackLevel</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>DesiredRedVideoBlackLevel</name>
          <direction>in</direction>
          <relatedStateVariable>RedVideoBlackLevel</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetGreenVideoBlackLevel</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>CurrentGreenVideoBlackLevel</name>
          <direction>out</direction>
          <relatedStateVariable>GreenVideoBlackLevel</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>SetGreenVideoBlackLevel</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>DesiredGreenVideoBlackLevel</name>
          <direction>in</direction>
          <relatedStateVariable>GreenVideoBlackLevel</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetBlueVideoBlackLevel</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>CurrentBlueVideoBlackLevel</name>
          <direction>out</direction>
          <relatedStateVariable>BlueVideoBlackLevel</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
```

```
    <action>
    <name>SetBlueVideoBlackLevel</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>DesiredBlueVideoBlackLevel</name>
    <direction>in</direction>
  <relatedStateVariable>BlueVideoBlackLevel</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetColorTemperature</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>CurrentColorTemperature</name>
          <direction>out</direction>
          <relatedStateVariable>ColorTemperature</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>SetColorTemperature</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>DesiredColorTemperature</name>
          <direction>in</direction>
          <relatedStateVariable>ColorTemperature</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetHorizontalKeystone</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>CurrentHorizontalKeystone</name>
          <direction>out</direction>
          <relatedStateVariable>HorizontalKeystone</relatedStateVariable>
        </argument>
      </argumentList>
```

```
    </action>
    <action>
    <name>SetHorizontalKeystone</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>DesiredHorizontalKeystone</name>
          <direction>in</direction>
          <relatedStateVariable>HorizontalKeystone</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetVerticalKeystone</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>CurrentVerticalKeystone</name>
          <direction>out</direction>
          <relatedStateVariable>VerticalKeystone</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>SetVerticalKeystone</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>DesiredVerticalKeystone</name>
          <direction>in</direction>
          <relatedStateVariable>VerticalKeystone</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetMute</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>Channel</name>
          <direction>in</direction>
          <relatedStateVariable>A_ARG_TYPE_Channel</relatedStateVariable>
        </argument>
```

```
        <argument>
          <name>CurrentMute</name>
          <direction>out</direction>
          <relatedStateVariable>Mute</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>SetMute</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>Channel</name>
          <direction>in</direction>
          <relatedStateVariable>A_ARG_TYPE_Channel</relatedStateVariable>
        </argument>
        <argument>
          <name>DesiredMute</name>
          <direction>in</direction>
          <relatedStateVariable>Mute</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetVolume</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>Channel</name>
          <direction>in</direction>
          <relatedStateVariable>A_ARG_TYPE_Channel</relatedStateVariable>
        </argument>
        <argument>
          <name>CurrentVolume</name>
          <direction>out</direction>
          <relatedStateVariable>Volume</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>SetVolume</name>
      <argumentList>
        <argument>
          <name>InstanceID</name>
          <direction>in</direction>

<relatedStateVariable>A_ARG_TYPE_InstanceID</relatedStateVariable>
        </argument>
        <argument>
          <name>Channel</name>
          <direction>in</direction>
          <relatedStateVariable>A_ARG_TYPE_Channel</relatedStateVariable>
        </argument>
```