

# INTERNATIONAL STANDARD



Information technology – UPnP device architecture –  
Part 4-13: Audio Video Device Control Protocol – Level 2 – Rendering Control  
Service

**ITih STANDARD PREVIEW**  
(standards.iteh.ai)

ISO/IEC 29341-4-13:2011  
<https://standards.iteh.ai/catalog/standards/sist/a6a60495-10dc-4737-bab0-d98517de949a/iso-iec-29341-4-13-2011>



**THIS PUBLICATION IS COPYRIGHT PROTECTED**  
**Copyright © 2011 ISO/IEC, Geneva, Switzerland**

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about ISO/IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

IEC Central Office  
3, rue de Varembe  
CH-1211 Geneva 20  
Switzerland  
Email: [inmail@iec.ch](mailto:inmail@iec.ch)  
Web: [www.iec.ch](http://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](http://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 It also gives information on projects, withdrawn and replaced publications.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](http://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](http://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](http://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](mailto:csc@iec.ch)  
Tel.: +41 22 919 02 11  
Fax: +41 22 919 03 00



ISO/IEC 29341-4-13

Edition 2.0 2011-09

# INTERNATIONAL STANDARD



Information technology – UPnP device architecture –  
Part 4-13: Audio Video Device Control Protocol – Level 2 – Rendering Control  
Service

<https://standards.iteh.ai/catalog/standards/sist/a6a60495-10dc-4737-bab0-d98517de949a/iso-iec-29341-4-13-2011>

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

PRICE CODE



ICS 35.200

ISBN 978-2-88912-682-8

## CONTENTS

1	Overview and Scope.....	6
1.1	Introduction .....	6
1.2	Multi-input Devices .....	6
1.3	Notation .....	6
1.3.1	Data Types .....	7
1.3.2	Strings Embedded in Other Strings.....	7
1.3.3	Extended Backus-Naur Form.....	8
1.4	Derived Data Types .....	8
1.4.1	Comma Separated Value (CSV) Lists .....	8
1.5	Management of XML Namespaces in Standardized DCPs.....	9
1.5.1	Namespace Prefix Requirements.....	12
1.5.2	Namespace Names, Namespace Versioning and Schema Versioning .....	13
1.5.3	Namespace Usage Examples .....	15
1.6	Vendor-defined Extensions .....	15
1.6.1	Vendor-defined Action Names .....	15
1.6.2	Vendor-defined State Variable Names .....	16
1.6.3	Vendor-defined XML Elements and attributes .....	16
1.6.4	Vendor-defined Property Names.....	16
1.7	References.....	16
2	Service Modeling Definitions.....	19
2.1	Service Type .....	19
2.2	State Variables.....	20
2.2.1	<u><a href="#">LastChange</a></u> .....	23
2.2.2	<u><a href="#">PresetNameList</a></u> .....	24
2.2.3	<u><a href="#">Brightness</a></u> .....	24
2.2.4	<u><a href="#">Contrast</a></u> .....	25
2.2.5	<u><a href="#">Sharpness</a></u> .....	25
2.2.6	<u><a href="#">RedVideoGain</a></u> .....	25
2.2.7	<u><a href="#">GreenVideoGain</a></u> .....	25
2.2.8	<u><a href="#">BlueVideoGain</a></u> .....	25
2.2.9	<u><a href="#">RedVideoBlackLevel</a></u> .....	25
2.2.10	<u><a href="#">GreenVideoBlackLevel</a></u> .....	25
2.2.11	<u><a href="#">BlueVideoBlackLevel</a></u> .....	26
2.2.12	<u><a href="#">ColorTemperature</a></u> .....	26
2.2.13	<u><a href="#">HorizontalKeystone</a></u> .....	26
2.2.14	<u><a href="#">VerticalKeystone</a></u> .....	26
2.2.15	<u><a href="#">Mute</a></u> .....	27
2.2.16	<u><a href="#">Volume</a></u> .....	27
2.2.17	<u><a href="#">VolumeDB</a></u> .....	27
2.2.18	<u><a href="#">Loudness</a></u> .....	28
2.2.19	<u><a href="#">A_ARG_TYPE_Channel</a></u> .....	28
2.2.20	<u><a href="#">A_ARG_TYPE_InstanceID</a></u> .....	29
2.2.21	<u><a href="#">A_ARG_TYPE_PresetName</a></u> .....	29
2.2.22	<u><a href="#">A_ARG_TYPE_DeviceUDN</a></u> .....	29
2.2.23	<u><a href="#">A_ARG_TYPE_ServiceType</a></u> .....	29

2.2.24	<u><a href="#">A_ARG_TYPE_ServiceID</a></u> .....	29
2.2.25	<u><a href="#">A_ARG_TYPE_StateVariableValuePairs</a></u> .....	29
2.2.26	<u><a href="#">A_ARG_TYPE_StateVariableList</a></u> .....	30
2.2.27	Relationships between State Variables.....	30
2.3	Eventing and Moderation.....	31
2.3.1	Event Model.....	32
2.4	Actions.....	32
2.4.1	<u><a href="#">ListPresets()</a></u> .....	34
2.4.2	<u><a href="#">SelectPreset()</a></u> .....	34
2.4.3	<u><a href="#">GetBrightness()</a></u> .....	35
2.4.4	<u><a href="#">SetBrightness()</a></u> .....	35
2.4.5	<u><a href="#">GetContrast()</a></u> .....	36
2.4.6	<u><a href="#">SetContrast()</a></u> .....	36
2.4.7	<u><a href="#">GetSharpness()</a></u> .....	37
2.4.8	<u><a href="#">SetSharpness()</a></u> .....	37
2.4.9	<u><a href="#">GetRedVideoGain()</a></u> .....	38
2.4.10	<u><a href="#">SetRedVideoGain()</a></u> .....	38
2.4.11	<u><a href="#">GetGreenVideoGain()</a></u> .....	39
2.4.12	<u><a href="#">SetGreenVideoGain()</a></u> .....	39
2.4.13	<u><a href="#">GetBlueVideoGain()</a></u> .....	40
2.4.14	<u><a href="#">SetBlueVideoGain()</a></u> .....	40
2.4.15	<u><a href="#">GetRedVideoBlackLevel()</a></u> .....	41
2.4.16	<u><a href="#">SetRedVideoBlackLevel()</a></u> .....	41
2.4.17	<u><a href="#">GetGreenVideoBlackLevel()</a></u> .....	42
2.4.18	<u><a href="#">SetGreenVideoBlackLevel()</a></u> .....	42
2.4.19	<u><a href="#">GetBlueVideoBlackLevel()</a></u> .....	43
2.4.20	<u><a href="#">SetBlueVideoBlackLevel()</a></u> .....	43
2.4.21	<u><a href="#">GetColorTemperature()</a></u> .....	44
2.4.22	<u><a href="#">SetColorTemperature()</a></u> .....	44
2.4.23	<u><a href="#">GetHorizontalKeystone()</a></u> .....	45
2.4.24	<u><a href="#">SetHorizontalKeystone()</a></u> .....	45
2.4.25	<u><a href="#">GetVerticalKeystone()</a></u> .....	46
2.4.26	<u><a href="#">SetVerticalKeystone()</a></u> .....	46
2.4.27	<u><a href="#">GetMute()</a></u> .....	47
2.4.28	<u><a href="#">SetMute()</a></u> .....	47
2.4.29	<u><a href="#">GetVolume()</a></u> .....	48
2.4.30	<u><a href="#">SetVolume()</a></u> .....	48
2.4.31	<u><a href="#">GetVolumeDB()</a></u> .....	49
2.4.32	<u><a href="#">SetVolumeDB()</a></u> .....	49
2.4.33	<u><a href="#">GetVolumeDBRange()</a></u> .....	50
2.4.34	<u><a href="#">GetLoudness()</a></u> .....	51
2.4.35	<u><a href="#">SetLoudness()</a></u> .....	51
2.4.36	<u><a href="#">GetStateVariables()</a></u> .....	52
2.4.37	<u><a href="#">SetStateVariables()</a></u> .....	53
2.4.38	Relationships Between Actions.....	53
2.4.39	Common Error Codes.....	54
2.5	Theory of Operation.....	54
2.5.1	Multi-input Devices.....	54
2.5.2	Presets.....	56

2.5.3	Controlling the Display of Visual Content .....	56
2.5.4	Controlling Audio Content .....	56
3	XML Service Description .....	58
4	Test .....	73
Figure 1:	Horizontal Keystone .....	26
Figure 2:	Vertical Keystone .....	27
Figure 3:	Relationship between <u>Volume</u> and <u>VolumeDB</u> .....	31
Figure 4:	Virtual Instances of RCS .....	55
Figure 5:	6-channel Volume Control .....	57
Table 1-1	— EBNF Operators .....	8
Table 1-2	— CSV Examples .....	9
Table 1-3	— Namespace Definitions .....	11
Table 1-4	— Schema-related Information .....	12
Table 1-5	— Default Namespaces for the AV Specifications .....	13
Table 2-1	— State Variables .....	20
Table 2-2	— allowedValueRange for <u>Brightness</u> .....	20
Table 2-3	— allowedValueRange for <u>Contrast</u> .....	21
Table 2-4	— allowedValueRange for <u>Sharpness</u> .....	21
Table 2-5	— allowedValueRange for <u>RedVideoGain</u> .....	21
Table 2-6	— allowedValueRange for <u>GreenVideoGain</u> .....	21
Table 2-7	— allowedValueRange for <u>BlueVideoGain</u> .....	21
Table 2-8	— allowedValueRange for <u>RedVideoBlackLevel</u> .....	21
Table 2-9	— allowedValueRange for <u>GreenVideoBlackLevel</u> .....	21
Table 2-10	— allowedValueRange for <u>BlueVideoBlackLevel</u> .....	21
Table 2-11	— allowedValueRange for <u>ColorTemperature</u> .....	22
Table 2-12	— allowedValueRange for <u>HorizontalKeystone</u> .....	22
Table 2-13	— allowedValueRange for <u>VerticalKeystone</u> .....	22
Table 2-14	— allowedValueRange for <u>Volume</u> .....	22
Table 2-15	— allowedValueRange for <u>VolumeDB</u> .....	22
Table 2-16	— allowedValueList for <u>A_ARG_TYPE_Channel</u> .....	23
Table 2-17	— allowedValueList for <u>A_ARG_TYPE_PresetName</u> .....	23
Table 2-18	— Predefined Names of Some Common Presets .....	29
Table 2-19	— Event moderation .....	31
Table 2-20	— Actions .....	33
Table 2-21	— Arguments for <u>ListPresets()</u> .....	34
Table 2-22	— Error Codes for <u>ListPresets()</u> .....	34
Table 2-23	— Arguments for <u>SelectPreset()</u> .....	34
Table 2-24	— Error Codes for <u>SelectPreset()</u> .....	35
Table 2-25	— Arguments for <u>GetBrightness()</u> .....	35
Table 2-26	— Error Codes for <u>GetBrightness()</u> .....	35
Table 2-27	— Arguments for <u>SetBrightness()</u> .....	35

Table 2-28 — Error Codes for <a href="#"><u>SetBrightness()</u></a> .....	36
Table 2-29 — Arguments for <a href="#"><u>GetContrast()</u></a> .....	36
Table 2-30 — Error Codes for <a href="#"><u>GetContrast()</u></a> .....	36
Table 2-31 — Arguments for <a href="#"><u>SetContrast()</u></a> .....	36
Table 2-32 — Error Codes for <a href="#"><u>SetContrast()</u></a> .....	37
Table 2-33 — Arguments for <a href="#"><u>GetSharpness()</u></a> .....	37
Table 2-34 — Error Codes for <a href="#"><u>GetSharpness()</u></a> .....	37
Table 2-35 — Arguments for <a href="#"><u>SetSharpness()</u></a> .....	37
Table 2-36 — Error Codes for <a href="#"><u>SetSharpness()</u></a> .....	38
Table 2-37 — Arguments for <a href="#"><u>GetRedVideoGain()</u></a> .....	38
Table 2-38 — Error Codes for <a href="#"><u>GetRedVideoGain()</u></a> .....	38
Table 2-39 — Arguments for <a href="#"><u>SetRedVideoGain()</u></a> .....	38
Table 2-40 — Error Codes for <a href="#"><u>SetRedVideoGain()</u></a> .....	39
Table 2-41 — Arguments for <a href="#"><u>GetGreenVideoGain()</u></a> .....	39
Table 2-42 — Error Codes for <a href="#"><u>GetGreenVideoGain()</u></a> .....	39
Table 2-43 — Arguments for <a href="#"><u>SetGreenVideoGain()</u></a> .....	39
Table 2-44 — Error Codes for <a href="#"><u>SetGreenVideoGain()</u></a> .....	40
Table 2-45 — Arguments for <a href="#"><u>GetBlueVideoGain()</u></a> .....	40
Table 2-46 — Error Codes for <a href="#"><u>GetBlueVideoGain()</u></a> .....	40
Table 2-47 — Arguments for <a href="#"><u>SetBlueVideoGain()</u></a> .....	40
Table 2-48 — Error Codes for <a href="#"><u>SetBlueVideoGain()</u></a> .....	41
Table 2-49 — Arguments for <a href="#"><u>GetRedVideoBlackLevel()</u></a> .....	41
Table 2-50 — Error Codes for <a href="#"><u>GetRedVideoBlackLevel()</u></a> .....	41
Table 2-51 — Arguments for <a href="#"><u>SetRedVideoBlackLevel()</u></a> .....	41
Table 2-52 — Error Codes for <a href="#"><u>SetRedVideoBlackLevel()</u></a> .....	42
Table 2-53 — Arguments for <a href="#"><u>GetGreenVideoBlackLevel()</u></a> .....	42
Table 2-54 — Error Codes for <a href="#"><u>GetGreenVideoBlackLevel()</u></a> .....	42
Table 2-55 — Arguments for <a href="#"><u>SetGreenVideoBlackLevel()</u></a> .....	42
Table 2-56 — Error Codes for <a href="#"><u>SetGreenVideoBlackLevel()</u></a> .....	43
Table 2-57 — Arguments for <a href="#"><u>GetBlueVideoBlackLevel()</u></a> .....	43
Table 2-58 — Error Codes for <a href="#"><u>GetBlueVideoBlackLevel()</u></a> .....	43
Table 2-59 — Arguments for <a href="#"><u>SetBlueVideoBlackLevel()</u></a> .....	43
Table 2-60 — Error Codes for <a href="#"><u>SetBlueVideoBlackLevel()</u></a> .....	44
Table 2-61 — Arguments for <a href="#"><u>GetColorTemperature()</u></a> .....	44
Table 2-62 — Error Codes for <a href="#"><u>GetColorTemperature()</u></a> .....	44
Table 2-63 — Arguments for <a href="#"><u>SetColorTemperature()</u></a> .....	44
Table 2-64 — Error Codes for <a href="#"><u>SetColorTemperature()</u></a> .....	45
Table 2-65 — Arguments for <a href="#"><u>GetHorizontalKeystone()</u></a> .....	45
Table 2-66 — Error Codes for <a href="#"><u>GetHorizontalKeystone()</u></a> .....	45
Table 2-67 — Arguments for <a href="#"><u>SetHorizontalKeystone()</u></a> .....	45
Table 2-68 — Error Codes for <a href="#"><u>SetHorizontalKeystone()</u></a> .....	46
Table 2-69 — Arguments for <a href="#"><u>GetVerticalKeystone()</u></a> .....	46
Table 2-70 — Error Codes for <a href="#"><u>GetVerticalKeystone()</u></a> .....	46

Table 2-71 — Arguments for <a href="#"><u>SetVerticalKeystone()</u></a> .....	46
Table 2-72 — Error Codes for <a href="#"><u>SetVerticalKeystone()</u></a> .....	47
Table 2-73 — Arguments for <a href="#"><u>GetMute()</u></a> .....	47
Table 2-74 — Error Codes for <a href="#"><u>GetMute()</u></a> .....	47
Table 2-75 — Arguments for <a href="#"><u>SetMute()</u></a> .....	47
Table 2-76 — Error Codes for <a href="#"><u>SetMute()</u></a> .....	48
Table 2-77 — Arguments for <a href="#"><u>GetVolume()</u></a> .....	48
Table 2-78 — Error Codes for <a href="#"><u>GetVolume()</u></a> .....	48
Table 2-79 — Arguments for <a href="#"><u>SetVolume()</u></a> .....	48
Table 2-80 — Error Codes for <a href="#"><u>SetVolume()</u></a> .....	49
Table 2-81 — Arguments for <a href="#"><u>GetVolumeDB()</u></a> .....	49
Table 2-82 — Error Codes for <a href="#"><u>GetVolumeDB()</u></a> .....	49
Table 2-83 — Arguments for <a href="#"><u>SetVolumeDB()</u></a> .....	50
Table 2-84 — Error Codes for <a href="#"><u>SetVolumeDB()</u></a> .....	50
Table 2-85 — Arguments for <a href="#"><u>GetVolumeDBRange()</u></a> .....	50
Table 2-86 — Error Codes for <a href="#"><u>GetVolumeDBRange()</u></a> .....	51
Table 2-87 — Arguments for <a href="#"><u>GetLoudness()</u></a> .....	51
Table 2-88 — Error Codes for <a href="#"><u>GetLoudness()</u></a> .....	51
Table 2-89 — Arguments for <a href="#"><u>SetLoudness()</u></a> .....	51
Table 2-90 — Error Codes for <a href="#"><u>SetLoudness()</u></a> .....	52
Table 2-91 — Arguments for <a href="#"><u>GetStateVariables()</u></a> .....	52
Table 2-92 — Error Codes for <a href="#"><u>GetStateVariables()</u></a> .....	52
Table 2-93 — Arguments for <a href="#"><u>SetStateVariables()</u></a> .....	53
Table 2-94 — Error Codes for <a href="#"><u>SetStateVariables()</u></a> .....	53
Table 2-95 — Common Error Codes .....	54

STANDARD PREVIEW

(standards.iteh.ai)

ISO/IEC 29341-4-13:2011

<https://standards.iteh.ai/catalog/standards/sist/60495-10dc-4737-bab0-29341-4-13-2011>



## INFORMATION TECHNOLOGY – UPNP DEVICE ARCHITECTURE –

### Part 4-13: Audio Video Device Control Protocol – Level 2 – 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.
- 10) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 29341-4-13 was prepared by UPnP Forum Steering committee<sup>1</sup>, was adopted, under the fast track procedure, by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

This International Standard replaces ISO/IEC 29341-4-13, first edition, published in 2008, and constitutes a technical revision.

The list of all currently available parts of the ISO/IEC 29341 series, under the general title *Information technology – UPnP device 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.

<sup>1</sup> UPnP Forum Steering committee, UPnP Forum, 3855 SW 153<sup>rd</sup> Drive, Beaverton, Oregon 97006 USA. See also "Introduction".

**IMPORTANT – The “colour inside” logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this publication using a colour printer.**

**iTeh STANDARD PREVIEW**  
**(standards.iteh.ai)**

[ISO/IEC 29341-4-13:2011](https://standards.iteh.ai/catalog/standards/sist/a6a60495-10dc-4737-bab0-d98517de949a/iso-iec-29341-4-13-2011)

<https://standards.iteh.ai/catalog/standards/sist/a6a60495-10dc-4737-bab0-d98517de949a/iso-iec-29341-4-13-2011>

## 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 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 service is intended to provide control points with the ability to query and/or adjust any rendering attribute that the device supports.

The RenderingControl 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 (that is: control) any modifiable attribute
- Restore the settings defined by a named Preset

The RenderingControl service DOES NOT:

- Control the flow of the associated content (for example, 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 (for example, 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 RenderingControl 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 RenderingControl service includes an *InstanceID* argument with each action that allows the control point to identify on which content the action is to be applied (for example, the post-mixed content or one of the pre-mixed input content items).

By convention, an *InstanceID* of 0 indicates that the invoked action MUST 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 Clause 2.5, "Theory of Operation" for additional information.

### 1.3 Notation

- In this document, features are described as Required, Recommended, or Optional as follows:

The keywords "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this specification are to be interpreted as described in [RFC 2119].

In addition, the following keywords are used in this specification:

PROHIBITED – The definition or behavior is prohibited by this specification. Opposite of REQUIRED.

**CONDITIONALLY REQUIRED** – The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is **REQUIRED**, otherwise it is **PROHIBITED**.

**CONDITIONALLY OPTIONAL** – The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is **OPTIONAL**, otherwise it is **PROHIBITED**.

These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

- Strings that are to be taken literally are enclosed in “double quotes”.
- Words that are emphasized are printed in *italic*.
- Keywords that are defined by the UPnP AV Working Committee are printed using the *forum* character style.
- Keywords that are defined by the UPnP Device Architecture specification are printed using the *arch* character style [DEVICE].
- A double colon delimiter, “::”, signifies a hierarchical parent-child (parent::child) relationship between the two objects separated by the double colon. This delimiter is used in multiple contexts, for example: Service::Action(), Action()::Argument, parentProperty::childProperty.

### 1.3.1 Data Types

This specification uses data type definitions from two different sources. The UPnP Device Architecture defined data types are used to define state variable and action argument data types [DEVICE]. The XML Schema namespace is used to define property data types [XML SCHEMA-2].

For UPnP Device Architecture defined **boolean** data types, it is strongly **RECOMMENDED** to use the value “0” for false, and the value “1” for true. However, when used as input arguments, the values “*false*”, “*no*”, “*true*”, “*yes*” may also be encountered and **MUST** be accepted. Nevertheless, it is strongly **RECOMMENDED** that all **boolean** state variables and output arguments be represented as “0” and “1”.

For XML Schema defined Boolean data types, it is strongly **RECOMMENDED** to use the value “0” for false, and the value “1” for true. However, when used as input properties, the values “*false*”, “*true*” may also be encountered and **MUST** be accepted. Nevertheless, it is strongly **RECOMMENDED** that all Boolean properties be represented as “0” and “1”.

### 1.3.2 Strings Embedded in Other Strings

Some string variables and arguments described in this document contain substrings that **MUST** be independently identifiable and extractable for other processing. This requires the definition of appropriate substring delimiters and an escaping mechanism so that these delimiters can also appear as ordinary characters in the string and/or its independent substrings. This document uses embedded strings in two contexts – Comma Separated Value (CSV) lists (see Clause 1.4.1, “Comma Separated Value (CSV) Lists”) and property values in search criteria strings. Escaping conventions use the backslash character, “\” (character code U+005C), as follows:

- a) Backslash (“\”) is represented as “\\” in both contexts.
- b) Comma (“,”) is
  - 1) represented as “\,” in individual substring entries in CSV lists
  - 2) not escaped in search strings

- c) Double quote (""") is
  - 1) not escaped in CSV lists
  - 2) not escaped in search strings when it appears as the start or end delimiter of a property value
  - 3) represented as \" in search strings when it appears as a character that is part of the property value

**1.3.3 Extended Backus-Naur Form**

Extended Backus-Naur Form is used in this document for a formal syntax description of certain constructs. The usage here is according to the reference [EBNF].

**1.3.3.1 Typographic conventions for EBNF**

Non-terminal symbols are unquoted sequences of characters from the set of English upper and lower case letters, the digits “0” through “9”, and the hyphen (“-”). Character sequences between 'single quotes' are terminal strings and MUST appear literally in valid strings. Character sequences between (\*comment delimiters\*) are English language definitions or supplementary explanations of their associated symbols. White space in the EBNF is used to separate elements of the EBNF, not to represent white space in valid strings. White space usage in valid strings is described explicitly in the EBNF. Finally, the EBNF uses the following operators:

**Table 1-1 — EBNF Operators**

Operator	Semantics
::=	<b>definition</b> – the non-terminal symbol on the left is defined by one or more alternative sequences of terminals and/or non-terminals to its right.
	<b>alternative separator</b> – separates sequences on the right that are independently allowed definitions for the non-terminal on the left.
*	<b>null repetition</b> – means the expression to its left MAY occur zero or more times.
+	<b>non-null repetition</b> – means the expression to its left MUST occur at least once and MAY occur more times.
[ ]	<b>optional</b> – the expression between the brackets is optional.
( )	<b>grouping</b> – groups the expressions between the parentheses.
-	<b>character range</b> – represents all characters between the left and right character operands inclusively.

**1.4 Derived Data Types**

This clause defines a derived data type that is represented as a string data type with special syntax. This specification uses string data type definitions that originate from two different sources. The UPnP Device Architecture defined string data type is used to define state variable and action argument string data types. The XML Schema namespace is used to define property xsd:string data types. The following definition applies to both string data types.

**1.4.1 Comma Separated Value (CSV) Lists**

The UPnP AV services use state variables, action arguments and properties that represent lists – or one-dimensional arrays – of values. The UPnP Device Architecture, Version 1.0 [DEVICE], does not provide for either an array type or a list type, so a list type is defined here. Lists MAY either be homogeneous (all values are the same type) or heterogeneous (values of different types are allowed). Lists MAY also consist of repeated occurrences of homogeneous or heterogeneous subsequences, all of which have the same syntax and semantics (same number of values, same value types and in the same order). The data type of a homogeneous list is string or xsd:string and denoted by CSV (x), where x is the type of the individual values. The data type of a heterogeneous list is also string or xsd:string and denoted by CSV (x, y, z), where x, y and z are the types of the individual values. If the number of values in the heterogeneous list is too large to show each type individually, that variable type is represented as CSV (heterogeneous), and the variable description includes additional information as to the expected sequence of values appearing in the list and their corresponding types. The data type of a repeated subsequence list is string or xsd:string and

denoted by CSV ({x, y, z}), where x, y and z are the types of the individual values in the subsequence and the subsequence MAY be repeated zero or more times.

- A list is represented as a **string** type (for state variables and action arguments) or xsd:string type (for properties).
- Commas separate values within a list.
- Integer values are represented in CSVs with the same syntax as the integer data type specified in [DEVICE] (that is: optional leading sign, optional leading zeroes, numeric US-ASCII)
- Boolean values are represented in state variable and action argument CSVs as either “**0**” for false or “**1**” for true. These values are a subset of the defined **boolean** data type values specified in [DEVICE]: **0, false, no, 1, true, yes.**
- Boolean values are represented in property CSVs as either “**0**” for false or “**1**” for true. These values are a subset of the defined Boolean data type values specified in [XML SCHEMA-2]: 0, false, 1, true.
- Escaping conventions for the comma and backslash characters are defined in Clause 1.3.2, “Strings Embedded in Other Strings”.
- White space before, after, or interior to any numeric data type is not allowed.
- White space before, after, or interior to any other data type is part of the value.

**Table 1-2 — CSV Examples**

Type refinement of string	Value	Comments
CSV ( <b>string</b> ) or CSV (xsd:string)	+artist,-date” (standards.iteh.ai)	List of 2 property sort criteria.
CSV ( <b>int</b> ) or CSV (xsd:integer)	“1,-5,006,0,+7” ISO/IEC 29341-4-13:2011	List of 5 integers.
CSV ( <b>boolean</b> ) or CSV (xsd:Boolean)	“0,1,1,0” https://standards.iteh.ai/catalog/standards/sist/a6a60495-10dc-4737-bab0-d98517de949a/iso-iec-29341-4-13-2011	List of 4 booleans
CSV ( <b>string</b> ) or CSV (xsd:string)	“Smith\, Fred,Jones\, Davey”	List of 2 names, “Smith, Fred” and “Jones, Davey”
CSV ( <b>i4,string,ui2</b> ) or CSV (xsd:int, xsd:string, xsd:unsignedShort)	“-29837, string with leading blanks,0”	Note that the second value is “ string with leading blanks”
CSV ( <b>i4</b> ) or CSV (xsd:int)	“3, 4”	Illegal CSV. White space is not allowed as part of an integer value.
CSV ( <b>string</b> ) or CSV (xsd:string)	“,,”	List of 3 empty string values
CSV (heterogeneous)	“Alice,Marketing,5,Sue,R&D,21,Dave,Finance,7”	List of unspecified number of people and associated attributes. Each person is described by 3 elements: a name <b>string</b> , a department <b>string</b> and years-of-service <b>ui2</b> or a name xsd:string, a department xsd:string and years-of-service xsd:unsignedShort.

### 1.5 Management of XML Namespaces in Standardized DCPs

UPnP specifications make extensive use of XML namespaces. This allows separate DCPs, and even separate components of an individual DCP, to be designed independently and still avoid name collisions when they share XML documents. Every name in an XML document belongs to exactly one namespace. In documents, XML names appear in one of two forms: qualified or unqualified. An unqualified name (or no-colon-name) contains no colon (“:”)

characters. An unqualified name belongs to the document's default namespace. A qualified name is two no-colon-names separated by one colon character. The no-colon-name before the colon is the qualified name's namespace prefix, the no-colon-name after the colon is the qualified name's "local" name (meaning local to the namespace identified by the namespace prefix). Similarly, the unqualified name is a local name in the default namespace.

The formal name of a namespace is a URI. The namespace prefix used in an XML document is *not* the name of the namespace. The namespace name is, or should be, globally unique. It has a single definition that is accessible to anyone who uses the namespace. It has the same meaning anywhere that it is used, both inside and outside XML documents. The namespace prefix, however, in formal XML usage, is defined only in an XML document. It must be locally unique to the document. Any valid XML no-colon-name may be used. And, in formal XML usage, no two XML documents are ever required to use the same namespace prefix to refer to the same namespace. The creation and use of the namespace prefix was standardized by the W3C XML Committee in [XML-NMSP] strictly as a convenient local shorthand replacement for the full URI name of a namespace in individual documents.

All AV object properties are represented in XML by element and attribute names, therefore, all property names belong to an XML namespace.

For the same reason that namespace prefixes are convenient in XML documents, it is convenient in specification text to refer to namespaces using a namespace prefix. Therefore, this specification declares a "standard" prefix for all XML namespaces used herein. In addition, this specification expands the scope where these prefixes have meaning, beyond a single XML document, to all of its text, XML examples, and certain string-valued properties. This expansion of scope does *not* supersede XML rules for usage in documents, it only augments and complements them in important contexts that are out-of-scope for the XML specifications. For example, action arguments which refer to CDS properties, such as the [SearchCriteria](#) argument of the [Search\(\)](#) action or the [Filter](#) argument of the [Browse\(\)](#) action, MUST use the predefined namespace prefixes when referring to CDS properties ("upnp:", "dc:", etc).

ISO/IEC 29341-4-13:2011

All of the namespaces used in this specification are listed in the Tables "Namespace Definitions" and "Schema-related Information". For each such namespace, Table 1-3, "Namespace Definitions" gives a brief description of it, its name (a URI) and its defined "standard" prefix name. Some namespaces included in these tables are not directly used or referenced in this document. They are included for completeness to accommodate those situations where this specification is used in conjunction with other UPnP specifications to construct a complete system of devices and services. For example, since the Scheduled Recording Service depends on and refers to the Content Directory Service, the predefined "srs:" namespace prefix is included. The individual specifications in such collections all use the same standard prefix. The standard prefixes are also used in Table 1-4, "Schema-related Information", to cross-reference additional namespace information. This second table includes each namespace's valid XML document root element(s) (if any), its schema file name, versioning information (to be discussed in more detail below), and a link to the entry in Clause 1.7, "References" for its associated schema.

The normative definitions for these namespaces are the documents referenced in Table 1-3. The schemas are designed to support these definitions for both human understanding and as test tools. However, limitations of the XML Schema language itself make it difficult for the UPnP-defined schemas to accurately represent all details of the namespace definitions. As a result, the schemas will validate many XML documents that are not valid according to the specifications.

The Working Committee expects to continue refining these schemas after specification release to reduce the number of documents that are validated by the schemas while violating the specifications, but the schemas will still be informative, supporting documents. Some schemas might become normative in future versions of the specifications.