



BSI Standards Publication

Information technology — UPnP Device Architecture

Part 20-13: Audio video device control protocol —
Level 4 — Rendering control service

National foreword

This British Standard is the UK implementation of ISO/IEC 29341-20-13:2017.

The UK participation in its preparation was entrusted to Technical Committee ICT/-/1, Information systems co-ordination.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2017
Published by BSI Standards Limited 2017

ISBN 978 0 580 90847 7

ICS 35.200

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 September 2017.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

**Information technology — UPnP
Device Architecture —**

**Part 20-13:
Audio video device control protocol —
Level 4 — Rendering control service**

Technologies de l'information — Architecture de dispositif UPnP —

*Partie 20-13: Protocole de contrôle de dispositif audio-vidéo —
Niveau 4 — Service de contrôle de rendu*



Reference number
ISO/IEC 29341-20-13:2017(E)

Documents
Document Sharing Hub
© ISO/IEC 2017



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2017, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

CONTENTS

1	Scope	1
1.1	Introduction	1
1.2	Multi-input Devices	1
2	Normative references	1
3	Terms, definitions, symbols and abbreviations	4
3.1	Provisioning terms	4
3.2	Symbols	5
4	Notations and Conventions	5
4.1	Notation	5
4.1.1	Data Types	5
4.1.2	Strings Embedded in Other Strings	6
4.1.3	Extended Backus-Naur Form	6
4.2	Derived Data Types	7
4.2.1	Summary	7
4.2.2	CSV Lists	7
4.3	Management of XML Namespaces in Standardized DCPs	8
4.3.1	Namespace Prefix Requirements	12
4.3.2	Namespace Names, Namespace Versioning and Schema Versioning	13
4.3.3	Namespace Usage Examples	15
4.4	Vendor-defined Extensions	15
4.4.1	Vendor-defined Action Names	15
4.4.2	Vendor-defined State Variable Names	15
4.4.3	Vendor-defined XML Elements and attributes	16
4.4.4	Vendor-defined Property Names	16
5	Service Modeling Definitions	16
5.1	Service Type	16
5.2	State Variables	17
5.2.1	State Variable Overview	17
5.2.2	<u>LastChange</u>	20
5.2.3	<u>PresetNameList</u>	23
5.2.4	<u>Brightness</u>	23
5.2.5	<u>Contrast</u>	23
5.2.6	<u>Sharpness</u>	23
5.2.7	<u>RedVideoGain</u>	23
5.2.8	<u>GreenVideoGain</u>	23
5.2.9	<u>BlueVideoGain</u>	24
5.2.10	<u>RedVideoBlackLevel</u>	24
5.2.11	<u>GreenVideoBlackLevel</u>	24
5.2.12	<u>BlueVideoBlackLevel</u>	24
5.2.13	<u>ColorTemperature</u>	24
5.2.14	<u>HorizontalKeystone</u>	24
5.2.15	<u>VerticalKeystone</u>	25
5.2.16	<u>Mute</u>	26
5.2.17	<u>Volume</u>	26
5.2.18	<u>VolumeDB</u>	26
5.2.19	<u>Loudness</u>	26

5.2.20	<u>AllowedTransformSettings</u>	27
5.2.21	<u>TransformSettings</u>	30
5.2.22	<u>AllowedDefaultTransformSettings</u>	31
5.2.23	<u>DefaultTransformSettings</u>	31
5.2.24	<u>A ARG TYPE Channel</u>	32
5.2.25	<u>A ARG TYPE InstanceID</u>	32
5.2.26	<u>A ARG TYPE PresetName</u>	32
5.2.27	<u>A ARG TYPE DeviceUDN</u>	32
5.2.28	<u>A ARG TYPE ServiceType</u>	32
5.2.29	<u>A ARG TYPE ServiceID</u>	33
5.2.30	<u>A ARG TYPE StateVariableValuePairs</u>	33
5.2.31	<u>A ARG TYPE StateVariableList</u>	33
5.2.32	Relationships between State Variables	33
5.3	Eventing and Moderation	34
5.3.1	Eventing and Moderation Overview	34
5.3.2	Event Model	36
5.4	Actions	36
5.4.1	Action Overview	36
5.4.2	<u>ListPresets()</u>	39
5.4.3	<u>SelectPreset()</u>	39
5.4.4	<u>GetBrightness()</u>	40
5.4.5	<u>SetBrightness()</u>	40
5.4.6	<u>GetContrast()</u>	41
5.4.7	<u>SetContrast()</u>	41
5.4.8	<u>GetSharpness()</u>	42
5.4.9	<u>SetSharpness()</u>	42
5.4.10	<u>GetRedVideoGain()</u>	43
5.4.11	<u>SetRedVideoGain()</u>	43
5.4.12	<u>GetGreenVideoGain()</u>	44
5.4.13	<u>SetGreenVideoGain()</u>	45
5.4.14	<u>GetBlueVideoGain()</u>	45
5.4.15	<u>SetBlueVideoGain()</u>	46
5.4.16	<u>GetRedVideoBlackLevel()</u>	46
5.4.17	<u>SetRedVideoBlackLevel()</u>	47
5.4.18	<u>GetGreenVideoBlackLevel()</u>	47
5.4.19	<u>SetGreenVideoBlackLevel()</u>	48
5.4.20	<u>GetBlueVideoBlackLevel()</u>	48
5.4.21	<u>SetBlueVideoBlackLevel()</u>	49
5.4.22	<u>GetColorTemperature()</u>	50
5.4.23	<u>SetColorTemperature()</u>	50
5.4.24	<u>GetHorizontalKeystone()</u>	51
5.4.25	<u>SetHorizontalKeystone()</u>	51
5.4.26	<u>GetVerticalKeystone()</u>	52
5.4.27	<u>SetVerticalKeystone()</u>	52
5.4.28	<u>GetMute()</u>	53
5.4.29	<u>SetMute()</u>	54
5.4.30	<u>GetVolume()</u>	54
5.4.31	<u>SetVolume()</u>	55
5.4.32	<u>GetVolumeDB()</u>	55

5.4.33	<u>SetVolumeDB()</u>	56
5.4.34	<u>GetVolumeDBRange()</u>	57
5.4.35	<u>GetLoudness()</u>	57
5.4.36	<u>SetLoudness()</u>	58
5.4.37	<u>GetStateVariables()</u>	59
5.4.38	<u>SetStateVariables()</u>	59
5.4.39	<u>GetAllowedTransforms()</u>	60
5.4.40	<u>GetTransforms()</u>	61
5.4.41	<u>SetTransforms()</u>	62
5.4.42	<u>GetAllowedDefaultTransforms()</u>	62
5.4.43	<u>GetDefaultTransforms()</u>	63
5.4.44	<u>SetDefaultTransforms()</u>	64
5.4.45	<u>GetAllAvailableTransforms()</u>	65
5.4.46	Relationships Between Actions	65
5.4.47	Common Error Codes	66
6	XML Service Description	66
7	Test	83
Annex A	(informative) Theory of Operation	84
A.1	Multi-input Devices	84
A.2	Presets	85
A.3	Controlling the Display of Visual Content	85
A.4	Controlling Audio Content	86
A.5	Transforms	88
A.5.1	Retrieving Transforms	88
A.5.2	Get Allowed Transforms from an instance	89
A.5.3	Setting Transforms	91
A.5.4	Retrieving Current values of the Transforms	91
A.5.5	Querying and setting default values for a Transform	92
Annex B	(normative) Pre-defined Transforms	94
B.1	Summary	94
B.2	<u>Rotation</u>	96
B.3	<u>RedEye</u>	96
B.4	<u>Zoom</u>	97
B.4.1	Additional units for the <u>Zoom</u> Transform	97
B.5	<u>HorizontalPan</u>	98
B.5.1	Additional units for the HorizontalPan Transform	98
B.6	<u>VerticalPan</u>	99
B.6.1	Additional units for the <u>VerticalPan</u> Transform	100
B.7	<u>ImageDisplayTime</u>	101
B.8	<u>ImageTransitionEffects</u>	102
B.9	<u>Equalization</u>	102
B.10	<u>BandEq [XX] [YY]</u>	103
B.10.1	Additional units for the <u>BandEq [XX] [YY]</u> Transform	104
B.11	<u>SpeakerConfiguration</u>	104
B.12	<u>OutputSelection [Name]</u>	105
B.13	<u>AudioTrackSelection</u>	106
B.14	<u>ClosedCaptioning</u>	107
B.15	<u>Subtitle</u>	108

B.16	<u>CameraAngle</u>	109
B.17	<u>PiP</u>	109
B.18	<u>ComponentInfoSelection</u>	110
B.19	<u>3DSettings3DScreen</u>	111
B.20	<u>3DSettings2DScreen</u>	113
B.21	Legacy compatible transforms	113
B.21.1	<u>Volume [Channel]</u>	113
B.21.2	<u>VolumeDB [Channel]</u>	114
B.21.3	<u>Mute [Channel]</u>	114
B.21.4	<u>Loudness [Channel]</u>	115
B.21.5	<u>Brightness</u>	116
B.21.6	<u>Sharpness</u>	116
B.21.7	<u>Contrast</u>	117
B.21.8	<u>RedVideoGain</u>	117
B.21.9	<u>GreenVideoGain</u>	117
B.21.10	<u>BlueVideoGain</u>	118
B.21.11	<u>RedVideoBlackLevel</u>	118
B.21.12	<u>GreenVideoBlackLevel</u>	119
B.21.13	<u>BlueVideoBlackLevel</u>	119
B.21.14	<u>ColorTemperature</u>	120
B.21.15	<u>HorizontalKeystone</u>	120
B.21.16	<u>VerticalKeystone</u>	121
Annex C (informative)	Bibliography	122

List of Tables

Table 1 — EBNF Operators.....	7
Table 2 — CSV Examples	8
Table 3 — Namespace Definitions.....	9
Table 4 — Schema-related Information.....	11
Table 5 — Default Namespaces for the AV Specifications	13
Table 6 — State Variables	17
Table 7 — allowedValueRange for <u>Brightness</u>	18
Table 8 — allowedValueRange for <u>Contrast</u>	18
Table 9 — allowedValueRange for <u>Sharpness</u>	18
Table 10 — allowedValueRange for <u>RedVideoGain</u>	18
Table 11 — allowedValueRange for <u>GreenVideoGain</u>	18
Table 12 — allowedValueRange for <u>BlueVideoGain</u>	18
Table 13 — allowedValueRange for <u>RedVideoBlackLevel</u>	19
Table 14 — allowedValueRange for <u>GreenVideoBlackLevel</u>	19
Table 15 — allowedValueRange for <u>BlueVideoBlackLevel</u>	19
Table 16 — allowedValueRange for <u>ColorTemperature</u>	19
Table 17 — allowedValueRange for <u>HorizontalKeystone</u>	19
Table 18 — allowedValueRange for <u>VerticalKeystone</u>	19
Table 19 — allowedValueRange for <u>Volume</u>	19
Table 20 — allowedValueRange for <u>VolumeDB</u>	19
Table 21 — allowedValueList for <u>A_ARG_TYPE_Channel</u>	20
Table 22 — allowedValueList for <u>A_ARG_TYPE_PresetName</u>	20
Table 23 — Allowed values for the <code>unit</code> attribute.....	29
Table 24 — Allowed values for the <code>scale</code> attribute.....	29
Table 25 — Predefined Names of Some Common Presets	32
Table 26 — Event moderation	35
Table 27 — Actions.....	37
Table 28 — Arguments for <u>ListPresets()</u>	39
Table 29 — Error Codes for <u>ListPresets()</u>	39
Table 30 — Arguments for <u>SelectPreset()</u>	39
Table 31 — Error Codes for <u>SelectPreset()</u>	40
Table 32 — Arguments for <u>GetBrightness()</u>	40
Table 33 — Error Codes for <u>GetBrightness()</u>	40
Table 34 — Arguments for <u>SetBrightness()</u>	40
Table 35 — Error Codes for <u>SetBrightness()</u>	41
Table 36 — Arguments for <u>GetContrast()</u>	41
Table 37 — Error Codes for <u>GetContrast()</u>	41
Table 38 — Arguments for <u>SetContrast()</u>	41
Table 39 — Error Codes for <u>SetContrast()</u>	42
Table 40 — Arguments for <u>GetSharpness()</u>	42
Table 41 — Error Codes for <u>GetSharpness()</u>	42

Table 42 — Arguments for <u>SetSharpness()</u>	43
Table 43 — Error Codes for <u>SetSharpness()</u>	43
Table 44 — Arguments for <u>GetRedVideoGain()</u>	43
Table 45 — Error Codes for <u>GetRedVideoGain()</u>	43
Table 46 — Arguments for <u>SetRedVideoGain()</u>	44
Table 47 — Error Codes for <u>SetRedVideoGain()</u>	44
Table 48 — Arguments for <u>GetGreenVideoGain()</u>	44
Table 49 — Error Codes for <u>GetGreenVideoGain()</u>	44
Table 50 — Arguments for <u>SetGreenVideoGain()</u>	45
Table 51 — Error Codes for <u>SetGreenVideoGain()</u>	45
Table 52 — Arguments for <u>GetBlueVideoGain()</u>	45
Table 53 — Error Codes for <u>GetBlueVideoGain()</u>	46
Table 54 — Arguments for <u>SetBlueVideoGain()</u>	46
Table 55 — Error Codes for <u>SetBlueVideoGain()</u>	46
Table 56 — Arguments for <u>GetRedVideoBlackLevel()</u>	46
Table 57 — Error Codes for <u>GetRedVideoBlackLevel()</u>	47
Table 58 — Arguments for <u>SetRedVideoBlackLevel()</u>	47
Table 59 — Error Codes for <u>SetRedVideoBlackLevel()</u>	47
Table 60 — Arguments for <u>GetGreenVideoBlackLevel()</u>	48
Table 61 — Error Codes for <u>GetGreenVideoBlackLevel()</u>	48
Table 62 — Arguments for <u>SetGreenVideoBlackLevel()</u>	48
Table 63 — Error Codes for <u>SetGreenVideoBlackLevel()</u>	48
Table 64 — Arguments for <u>GetBlueVideoBlackLevel()</u>	49
Table 65 — Error Codes for <u>GetBlueVideoBlackLevel()</u>	49
Table 66 — Arguments for <u>SetBlueVideoBlackLevel()</u>	49
Table 67 — Error Codes for <u>SetBlueVideoBlackLevel()</u>	50
Table 68 — Arguments for <u>GetColorTemperature()</u>	50
Table 69 — Error Codes for <u>GetColorTemperature()</u>	50
Table 70 — Arguments for <u>SetColorTemperature()</u>	50
Table 71 — Error Codes for <u>SetColorTemperature()</u>	51
Table 72 — Arguments for <u>GetHorizontalKeystone()</u>	51
Table 73 — Error Codes for <u>GetHorizontalKeystone()</u>	51
Table 74 — Arguments for <u>SetHorizontalKeystone()</u>	52
Table 75 — Error Codes for <u>SetHorizontalKeystone()</u>	52
Table 76 — Arguments for <u>GetVerticalKeystone()</u>	52
Table 77 — Error Codes for <u>GetVerticalKeystone()</u>	52
Table 78 — Arguments for <u>SetVerticalKeystone()</u>	53
Table 79 — Error Codes for <u>SetVerticalKeystone()</u>	53
Table 80 — Arguments for <u>GetMute()</u>	53
Table 81 — Error Codes for <u>GetMute()</u>	53
Table 82 — Arguments for <u>SetMute()</u>	54
Table 83 — Error Codes for <u>SetMute()</u>	54
Table 84 — Arguments for <u>GetVolume()</u>	54

Table 85 — Error Codes for <u>GetVolume()</u>	55
Table 86 — Arguments for <u>SetVolume()</u>	55
Table 87 — Error Codes for <u>SetVolume()</u>	55
Table 88 — Arguments for <u>GetVolumeDB()</u>	56
Table 89 — Error Codes for <u>GetVolumeDB()</u>	56
Table 90 — Arguments for <u>SetVolumeDB()</u>	56
Table 91 — Error Codes for <u>SetVolumeDB()</u>	57
Table 92 — Arguments for <u>GetVolumeDBRange()</u>	57
Table 93 — Error Codes for <u>GetVolumeDBRange()</u>	57
Table 94 — Arguments for <u>GetLoudness()</u>	58
Table 95 — Error Codes for <u>GetLoudness()</u>	58
Table 96 — Arguments for <u>SetLoudness()</u>	58
Table 97 — Error Codes for <u>SetLoudness()</u>	58
Table 98 — Arguments for <u>GetStateVariables()</u>	59
Table 99 — Error Codes for <u>GetStateVariables()</u>	59
Table 100 — Arguments for <u>SetStateVariables()</u>	60
Table 101 — Error Codes for <u>SetStateVariables()</u>	60
Table 102 — Arguments for <u>GetAllowedTransforms()</u>	61
Table 103 — Error Codes for <u>GetAllowedTransforms()</u>	61
Table 104 — Arguments for <u>GetTransforms()</u>	61
Table 105 — Error Codes for <u>GetTransforms()</u>	61
Table 106 — Arguments for <u>SetTransforms()</u>	62
Table 107 — Error Codes for <u>SetTransforms()</u>	62
Table 108 — Arguments for <u>GetAllowedDefaultTransforms()</u>	63
Table 109 — Error Codes for <u>GetAllowedDefaultTransforms()</u>	63
Table 110 — Arguments for <u>GetDefaultTransforms()</u>	63
Table 111 — Error Codes for <u>GetDefaultTransforms()</u>	64
Table 112 — Arguments for <u>SetDefaultTransforms()</u>	64
Table 113 — Error Codes for <u>SetDefaultTransforms()</u>	65
Table 114 — Arguments for <u>GetAllAvailableTransforms()</u>	65
Table 115 — Error Codes for <u>GetAllAvailableTransforms()</u>	65
Table 116 — Common Error Codes	66
Table B.1 — Pre-defined Transforms	95
Table B.2 — Recommended properties for <u>Rotation</u>	96
Table B.3 — allowedValueRange for <u>Rotation</u>	96
Table B.4 — Recommended properties for <u>RedEye</u>	96
Table B.5 — allowedValueList for <u>RedEye</u>	96
Table B.6 — Recommended properties for <u>Zoom</u>	97
Table B.7 — allowedValueRange for <u>Zoom</u>	97
Table B.8 — Alternative properties for <u>Zoom</u>	98
Table B.9 — allowedValueList for <u>Zoom</u> pre-defined values	98
Table B.10 — Recommended properties for <u>HorizontalPan</u>	98
Table B.11 — allowedValueRange for <u>HorizontalPan</u>	98

Table B.12 — Alternative properties for HorizontalPan (percentage unit)	99
Table B.13 — allowedValueRange for HorizontalPan (percentage unit)	99
Table B.14 — Alternative properties mm for HorizontalPan	99
Table B.15 — allowedValueRange for HorizontalPan (mm)	99
Table B.16 — Recommended properties for VerticalPan	100
Table B.17 — allowedValueRange for VerticalPan	100
Table B.18 — Alternative properties for VerticalPan (percentage unit)	100
Table B.19 — allowedValueRange for VerticalPan (percentage unit)	101
Table B.20 — Alternative properties for VerticalPan (mm unit)	101
Table B.21 — allowedValueRange for VerticalPan (mm unit)	101
Table B.22 — Recommended properties for ImageDisplayTime	101
Table B.23 — allowedValueRange for ImageDisplayTime (sec unit)	102
Table B.25 — allowedValueList for ImageTransitionEffects	102
Table B.26 — Recommended properties for Equalization	102
Table B.27 — allowedValueList for Equalization	103
Table B.28 — Recommended properties for BandEq [XX] [YY]	104
Table B.29 — allowedValueRange for each BandEq [XX] [YY]	104
Table B.30 — Alternative properties for BandEq [XX] [YY] (Equalization volume unit)	104
Table B.31 — allowedValueRange for BandEq [XX] [YY] (Equalization volume unit)	104
Table B.32 — Recommended properties for SpeakerConfiguration	105
Table B.33 — allowedValueList for SpeakerConfiguration	105
Table B.34 — Transform names based on OutputSelection [Name]	105
Table B.35 — Recommended properties for OutputSelection [Name]	106
Table B.36 — allowedValueList for OutputSelection [Name]	106
Table B.37 — Recommended properties for AudioTrackSelection	106
Table B.38 — allowedValueList for AudioTrackSelection (language mode)	106
Table B.39 — Alternative properties for AudioTrackSelection	107
Table B.40 — allowedValueList for AudioTrackSelection (track indication mode)	107
Table B.41 — Recommended properties for ClosedCaptioning	107
Table B.42 — allowedValueList for ClosedCaptioning	107
Table B.43 — Alternative properties for ClosedCaptioning	108
Table B.44 — allowedValueList for ClosedCaptioning (indication mode)	108
Table B.45 — Recommended properties for Subtitle	108
Table B.46 — allowedValueList for Subtitle	108
Table B.47 — Alternative properties for Subtitle	109
Table B.48 — allowedValueList for Subtitle (track indication mode)	109
Table B.49 — Recommended properties for CameraAngle	109
Table B.50 — allowedValueList for CameraAngle	109
Table B.51 — Recommended properties for PiP	110
Table B.52 — allowedValueList for PiP	110
Table B.53 — Recommended properties for ComponentInfoSelection	110
Table B.54 — allowedValueList for ComponentInfoSelection	110
Table B.55 — Recommended properties for 3DSettings3DScreen	111

Table B.56 — allowedValueList for <u>3DSettings3DScreen</u>	112
Table B.57 — Recommended properties <u>3DSettings2DScreen</u>	113
Table B.58 — allowedValueList for <u>3DSettings2DScreen</u>	113
Table B.59 — Recommended properties for <u>Volume [Channel]</u>	114
Table B.60 — allowedValueRange for <u>Volume [Channel]</u>	114
Table B.61 — Recommended properties for <u>VolumeDB [Channel]</u>	114
Table B.62 — allowedValueRange for <u>VolumeDB [Channel]</u>	114
Table B.63 — Recommended properties for <u>Mute [Channel]</u>	115
Table B.64 — allowedValueList for <u>Mute [Channel]</u>	115
Table B.65 — Recommended properties for <u>Loudness [Channel]</u>	115
Table B.66 — allowedValueList for <u>Loudness [Channel]</u>	116
Table B.67 — Recommended properties for <u>Brightness</u>	116
Table B.68 — allowedValueRange for <u>Brightness</u>	116
Table B.69 — Recommended properties for <u>Sharpness</u>	116
Table B.70 — allowedValueRange for <u>Sharpness</u>	116
Table B.71 — Recommended properties for <u>Contrast</u>	117
Table B.72 — allowedValueRange for <u>Contrast</u>	117
Table B.73 — Recommended properties for <u>RedVideoGain</u>	117
Table B.74 — allowedValueRange for <u>RedVideoGain</u>	117
Table B.75 — Recommended properties for <u>GreenVideoGain</u>	118
Table B.76 — allowedValueRange for <u>GreenVideoGain</u>	118
Table B.77 — Recommended properties for <u>BlueVideoGain</u>	118
Table B.78 — allowedValueRange for <u>BlueVideoGain</u>	118
Table B.79 — Recommended properties for <u>RedVideoBlackLevel</u>	119
Table B.80 — allowedValueRange for <u>RedVideoBlackLevel</u>	119
Table B.81 — Recommended properties for <u>GreenVideoBlackLevel</u>	119
Table B.82 — allowedValueRange for <u>GreenVideoBlackLevel</u>	119
Table B.83 — Recommended properties for <u>BlueVideoBlackLevel</u>	120
Table B.84 — allowedValueRange for <u>BlueVideoBlackLevel</u>	120
Table B.85 — Recommended properties for <u>ColorTemperature</u>	120
Table B.86 — allowedValueRange for <u>ColorTemperature</u>	120
Table B.87 — Recommended properties for <u>HorizontalKeystone</u>	121
Table B.88 — allowedValueRange for <u>HorizontalKeystone</u>	121
Table B.89 — Recommended properties for <u>VerticalKeystone</u>	121
Table B.90 — allowedValueRange for <u>VerticalKeystone</u>	121

List of Figures

Figure 1 — Horizontal Keystone	25
Figure 2 — Vertical Keystone	26
Figure 3 — Relationship between <u>Volume</u> and <u>VolumeDB</u>	34
Figure 4 — Virtual Instances of RCS	84
Figure 5 — 6-channel Volume Control	86
Figure 6 — Graphical presentation of an implementation of the <u>3D-2D-TopBottom</u> transform, in this case showing only the Top image of a 3D stream as a 2D image on the 2D display.....	111
Figure 7 — Graphical presentation of an implementation of the <u>3D-2D-SideBySide</u> transform, in this case showing only the Right image of a 3D stream as a 2D image on the 2D display.....	111
Figure 8 — Graphical presentation of an implementation of the <u>3D-3D-TopBottom</u> transform.....	112
Figure 9 — Graphical presentation of an implementation of the <u>3D-3D-SideBySide</u> transform.....	112

Foreword

ISO (the International Organization for Standardization) and IEC (the 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 through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see <http://www.iso.org/directives>).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of Standard, the meaning of the ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: [Foreword – Supplementary information](#)

ISO/IEC 29341-20-13 was prepared by UPnP Forum 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 ISO/IEC 29341 series, under the general title *Information technology — UPnP Device Architecture*, can be found on the [ISO web site](#).

Introduction

ISO and IEC 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 these patent rights. The holders of these patent rights have assured ISO and IEC that they are willing to negotiate licenses under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with ISO and IEC.

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

Huawei Technologies Co., Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Huawei Technologies Co., Ltd.
Administration Building, Bantian Longgang District
Shenzhen – China 518129

Qualcomm Incorporated has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Qualcomm Incorporated
5775 Morehouse Drive
San Diego, CA – USA 92121

Telecom Italia S.p.A. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Telecom Italia S.p.A.
Via Reiss Romoli, 274
Turin - Italy 10148

Cisco Systems informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA – USA 95134

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. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Original UPnP Document

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 and later by UPnP Forum. 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 29341-1:2008
UPnP Device Architecture Version 1.0	ISO/IEC 29341-1:2011
UPnP Device Architecture 1.1	ISO/IEC 29341-1-1:2011
UPnP Device Architecture 2.0	ISO/IEC 29341-1-2
UPnP Basic:1 Device	ISO/IEC 29341-2
UPnP AV Architecture:1	ISO/IEC 29341-3-1:2008
UPnP AV Architecture:1	ISO/IEC 29341-3-1:2011
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:1 Device	ISO/IEC 29341-3-2
UPnP MediaRenderer:2 Device	ISO/IEC 29341-3-2:2011
UPnP MediaServer:1 Device	ISO/IEC 29341-3-3
UPnP AVTransport:2 Service	ISO/IEC 29341-4-10:2008
UPnP AVTransport:2 Service	ISO/IEC 29341-4-10:2011
UPnP ConnectionManager:2 Service	ISO/IEC 29341-4-11:2008
UPnP ConnectionManager:2 Service	ISO/IEC 29341-4-11:2011
UPnP ContentDirectory:2 Service	ISO/IEC 29341-4-12
UPnP RenderingControl:2 Service	ISO/IEC 29341-4-13:2008
UPnP RenderingControl:2 Service	ISO/IEC 29341-4-13:2011
UPnP ScheduledRecording:1	ISO/IEC 29341-4-14
UPnP ScheduledRecording:2	ISO/IEC 29341-4-14:2011
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 29341-4-4:2008
UPnP AV Datastructure Template:1	ISO/IEC 29341-4-4:2011
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 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 HVAC_ZoneThermostat:1 Device	ISO/IEC 29341-6-2

UPnP BinaryLight:1 Device	ISO/IEC 29341-7-1
UPnP Dimming:1 Service	ISO/IEC 29341-7-10
UPnP SwitchPower:1 Service	ISO/IEC 29341-7-11
UPnP DimmableLight:1 Device	ISO/IEC 29341-7-2
UPnP InternetGatewayDevice:1 Device	ISO/IEC 29341-8-1
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 LANDevice:1 Device	ISO/IEC 29341-8-2
UPnP WANPPPConnection:1 Service	ISO/IEC 29341-8-20
UPnP WLANConfiguration:1 Service	ISO/IEC 29341-8-21
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 Printer:1 Device	ISO/IEC 29341-9-1
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 Scanner:1.0 Device	ISO/IEC 29341-9-2
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 QosDevice:2 Service	ISO/IEC 29341-11-10
UPnP QosManager:2 Service	ISO/IEC 29341-11-11
UPnP QosPolicyHolder:2 Service	ISO/IEC 29341-11-12
UPnP QOS v2 Schema Files	ISO/IEC 29341-11-2
UPnP RemoteUIClientDevice:1 Device	ISO/IEC 29341-12-1
UPnP RemoteUIClient:1 Service	ISO/IEC 29341-12-10
UPnP RemoteUIServer:1 Service	ISO/IEC 29341-12-11
UPnP RemoteUIServerDevice:1 Device	ISO/IEC 29341-12-2
UPnP DeviceSecurity:1 Service	ISO/IEC 29341-13-10
UPnP SecurityConsole:1 Service	ISO/IEC 29341-13-11
UPnP ContentDirectory:3 Service	ISO/IEC 29341-14-12:2011
UPnP MediaServer:3 Device	ISO/IEC 29341-14-3:2011
UPnP ContentSync:1	ISO/IEC 29341-15-10:2011
UPnP Low Power Architecture:1	ISO/IEC 29341-16-1:2011
UPnP LowPowerProxy:1 Service	ISO/IEC 29341-16-10:2011

UPnP LowPowerDevice:1 Service	ISO/IEC 29341-16-11:2011
UPnP QoS Architecture:3	ISO/IEC 29341-17-1:2011
UPnP QosDevice:3 Service	ISO/IEC 29341-17-10:2011
UPnP QosManager:3 Service	ISO/IEC 29341-17-11:2011
UPnP QosPolicyHolder:3 Service	ISO/IEC 29341-17-12:2011
UPnP QosDevice:3 Addendum	ISO/IEC 29341-17-13:2011
UPnP RemoteAccessArchitecture:1	ISO/IEC 29341-18-1:2011
UPnP InboundConnectionConfig:1 Service	ISO/IEC 29341-18-10:2011
UPnP RADAConfig:1 Service	ISO/IEC 29341-18-11:2011
UPnP RADASync:1 Service	ISO/IEC 29341-18-12:2011
UPnP RATAConfig:1 Service	ISO/IEC 29341-18-13:2011
UPnP RAClient:1 Device	ISO/IEC 29341-18-2:2011
UPnP RAServer:1 Device	ISO/IEC 29341-18-3:2011
UPnP RADiscoveryAgent:1 Device	ISO/IEC 29341-18-4:2011
UPnP SolarProtectionBlind:1 Device	ISO/IEC 29341-19-1:2011
UPnP TwoWayMotionMotor:1 Service	ISO/IEC 29341-19-10:2011
UPnP AV Architecture:2	ISO/IEC 29341-20-1
UPnP AVTransport:3 Service	ISO/IEC 29341-20-10
UPnP ConnectionManager:3 Service	ISO/IEC 29341-20-11
UPnP ContentDirectory:4 Device	ISO/IEC 29341-20-12
UPnP RenderingControl:3 Service	ISO/IEC 29341-20-13
UPnP ScheduledRecording:2 Service	ISO/IEC 29341-20-14
UPnP MediaRenderer:3 Service	ISO/IEC 29341-20-2
UPnP MediaServer:4 Device	ISO/IEC 29341-20-3
UPnP AV Datastructure Template:1	ISO/IEC 29341-20-4
UPnP InternetGatewayDevice:2 Device	ISO/IEC 29341-24-1
UPnP WANIPConnection:2 Service	ISO/IEC 29341-24-10
UPnP WANIPv6FirewallControl:1 Service	ISO/IEC 29341-24-11
UPnP WANConnectionDevice:2 Service	ISO/IEC 29341-24-2
UPnP WANDevice:2 Device	ISO/IEC 29341-24-3
UPnP Telephony Architecture:2	ISO/IEC 29341-26-1
UPnP CallManagement:2 Service	ISO/IEC 29341-26-10
UPnP MediaManagement:2 Service	ISO/IEC 29341-26-11
UPnP Messaging:2 Service	ISO/IEC 29341-26-12
UPnP PhoneManagement:2 Service	ISO/IEC 29341-26-13
UPnP AddressBook:1 Service	ISO/IEC 29341-26-14
UPnP Calendar:1 Service	ISO/IEC 29341-26-15
UPnP Presense:1 Service	ISO/IEC 29341-26-16
UPnP TelephonyClient:2 Device	ISO/IEC 29341-26-2
UPnP TelephonyServer:2 Device	ISO/IEC 29341-26-3
UPnP Friendly Info Update:1 Service	ISO/IEC 29341-27-1
UPnP MultiScreen MultiScreen Architecture:1	ISO/IEC 29341-28-1
UPnP MultiScreen Application Management:1 Service	ISO/IEC 29341-28-10
UPnP MultiScreen Screen:1 Device	ISO/IEC 29341-28-2
UPnP MultiScreen Application Management:2 Service	ISO/IEC 29341-29-10
UPnP MultiScreen Screen:2 Device	ISO/IEC 29341-29-2
UPnP IoT Management and Control Architecture Overview:1	ISO/IEC 29341-30-1

UPnP DataStore:1 Service	ISO/IEC 29341-30-10
UPnP IoT Management and Control Data Model:1 Service	ISO/IEC 29341-30-11
UPnP IoT Management and Control Transport Generic:1 Service	ISO/IEC 29341-30-12
UPnP IoT Management and Control:1 Device	ISO/IEC 29341-30-2
UPnP Energy Management:1 Service	ISO/IEC 29341-31-1

1 Scope

This service template is compliant with the UPnP Device Architecture version 1.0 [14]. 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:

- a) Discover the set of attributes supported by the device.
- b) Retrieve the current setting of any supported attribute
- c) Change the setting of (that is: control) any modifiable attribute
- d) Perform a set of content transforms, which, in addition to the above, also enables functionality for selecting content depended options, for example:
 - 1) Selecting a specific sub-stream in a composite stream for rendering.
 - 2) Turning subtitling on or off.
- e) Restore the settings defined by a named Preset

The RenderingControl service does not:

- a) Control the flow of the associated content (for example, Play, Stop, Pause, Seek, etc.).
- b) Provide a mechanism to enumerate locally stored content.
- c) 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 shall 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 Annex A for additional information.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

- [1] – *XML Schema for RenderingControl AllowedTransformSettings*, UPnP Forum, March 31, 2013.
Available at: <http://www.upnp.org/schemas/av/AllowedTransformSettings-v1-20130331.xsd>.
Latest version available at: <http://www.upnp.org/schemas/av/AllowedTransformSettings.xsd>.
- [2] – *AV Datastructure Template:1*, UPnP Forum, March 31, 2013.
Available at: <http://www.upnp.org/specs/av/UPnP-av-AVDataStructureTemplate-v1-20130331.pdf>.
Latest version available at: <http://www.upnp.org/specs/av/UPnP-av-AVDataStructureTemplate-v1.pdf>.
- [3] – *XML Schema for UPnP AV Common XML Data Types*, UPnP Forum, March 31, 2013.
Available at: <http://www.upnp.org/schemas/av/av-v3-20130331.xsd>.
Latest version available at: <http://www.upnp.org/schemas/av/av.xsd>.
- [4] – *XML Schema for UPnP AV Common XML Structures*, UPnP Forum, March 31, 2013.
Available at: <http://www.upnp.org/schemas/av/avs-v3-20130331.xsd>.
Latest version available at: <http://www.upnp.org/schemas/av/avs.xsd>.
- [5] – *AVTransport:3*, UPnP Forum, March 31, 2013.
Available at: <http://www.upnp.org/specs/av/UPnP-av-AVTransport-v3-Service-20130331.pdf>.
Latest version available at: <http://www.upnp.org/specs/av/UPnP-av-AVTransport-v3-Service.pdf>.
- [6] – *XML Schema for AVTransport LastChange Eventing*, UPnP Forum, September 30, 2008.
Available at: <http://www.upnp.org/schemas/av/avt-event-v2-20080930.xsd>.
Latest version available at: <http://www.upnp.org/schemas/av/avt-event.xsd>.
- [7] – *ContentDirectory:4*, UPnP Forum, March 31, 2013.
Available at: <http://www.upnp.org/specs/av/UPnP-av-ContentDirectory-v4-Service-20130331.pdf>.
Latest version available at: <http://www.upnp.org/specs/av/UPnP-av-ContentDirectory-v4-Service.pdf>.
- [8] – *XML Schema for ContentDirectory LastChange Eventing*, UPnP Forum, September 30, 2008.
Available at: <http://www.upnp.org/schemas/av/cds-event-v1-20080930.xsd>.
Latest version available at: <http://www.upnp.org/schemas/av/cds-event.xsd>.
- [9] – *ConnectionManager:3*, UPnP Forum, March 31, 2013.
Available at: <http://www.upnp.org/specs/av/UPnP-av-ConnectionManager-v3-Service-20130331.pdf>.
Latest version available at: <http://www.upnp.org/specs/av/UPnP-av-ConnectionManager-v3-Service.pdf>.
- [10] – *XML Schema for ConnectionManager DeviceClockInfoUpdates*, UPnP Forum, December 31, 2010.
Available at: <http://www.upnp.org/schemas/av/cm-deviceClockInfoUpdates-v1-20101231.xsd>.
Latest version available at: <http://www.upnp.org/schemas/av/cm-deviceClockInfoUpdates.xsd>.
- [11] – *XML Schema for ConnectionManager Features*, UPnP Forum, December 31, 2010.
Available at: <http://www.upnp.org/schemas/av/cm-featureList-v1-20101231.xsd>.
Latest version available at: <http://www.upnp.org/schemas/av/cm-featureList.xsd>.
- [12] – *XML Schema for UPnP AV Dublin Core*.
Available at: <http://www.dublincore.org/schemas/xmls/simpledc20020312.xsd>.
- [13] – *DCMI term declarations represented in XML schema language*.
Available at: <http://www.dublincore.org/schemas/xmls>.
- [14] – *UPnP Device Architecture, version 1.0*, UPnP Forum, October 15, 2008.
Available at: <http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0-20081015.pdf>.

Latest version available at: <http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0.pdf>.

[15] – *XML Schema for ContentDirectory Structure and Metadata (DIDL-Lite)*, UPnP Forum, March 31, 2013.

Available at: <http://www.upnp.org/schemas/av/didl-lite-v3-20130331.xsd>.

Latest version available at: <http://www.upnp.org/schemas/av/didl-lite.xsd>.

[16] – *XML Schema for ContentDirectory DeviceMode*, UPnP Forum, December 31, 2010.

Available at: <http://www.upnp.org/schemas/av/dmo-v1-20101231.xsd>.

Latest version available at: <http://www.upnp.org/schemas/av/dmo.xsd>.

[17] – *XML Schema for ContentDirectory DeviceModeRequest*, UPnP Forum, December 31, 2010.

Available at: <http://www.upnp.org/schemas/av/dmor-v1-20101231.xsd>.

Latest version available at: <http://www.upnp.org/schemas/av/dmor.xsd>.

[18] – *XML Schema for ContentDirectory DeviceModeStatus*, UPnP Forum, December 31, 2010.

Available at: <http://www.upnp.org/schemas/av/dmos-v1-20101231.xsd>.

Latest version available at: <http://www.upnp.org/schemas/av/dmos.xsd>.

[19] – ISO/IEC 14977, *Information technology - Syntactic metalanguage - Extended BNF*, December 1996.

[20] – *XML Schema for ContentDirectory PermissionsInfo*, UPnP Forum, December 31, 2010.

Available at: <http://www.upnp.org/schemas/av/pi-v1-20101231.xsd>.

Latest version available at: <http://www.upnp.org/schemas/av/pi.xsd>.

[21] – *RenderingControl:3*, UPnP Forum, March 31, 2013.

Available at: <http://www.upnp.org/specs/av/UPnP-av-RenderingControl-v3-Service-20130331.pdf>.

Latest version available at: <http://www.upnp.org/specs/av/UPnP-av-RenderingControl-v3-Service.pdf>.

[22] – *XML Schema for RenderingControl LastChange Eventing*, UPnP Forum, December 31, 2010.

Available at: <http://www.upnp.org/schemas/av/rcs-event-v3-20101231.xsd>.

Latest version available at: <http://www.upnp.org/schemas/av/rcs-event.xsd>.

[23] – *XML Schema for ConnectionManager RendererInfo*, UPnP Forum, December 31, 2010.

Available at: <http://www.upnp.org/schemas/av/rii-v1-20101231.xsd>.

Latest version available at: <http://www.upnp.org/schemas/av/rii.xsd>.

[24] – *XML Schema for AVTransport PlaylistInfo*, UPnP Forum, March 31, 2013.

Available at: <http://www.upnp.org/schemas/av/rpl-v1-20130331.xsd>.

Latest version available at: <http://www.upnp.org/schemas/av/rpl.xsd>.

[25] – *ScheduledRecording:2*, UPnP Forum, March 31, 2013.

Available at: <http://www.upnp.org/specs/av/UPnP-av-ScheduledRecording-v2-Service-20130331.pdf>.

Latest version available at: <http://www.upnp.org/specs/av/UPnP-av-ScheduledRecording-v2-Service.pdf>.

[26] – *XML Schema for ScheduledRecording Metadata and Structure*, UPnP Forum, March 31, 2013.

Available at: <http://www.upnp.org/schemas/av/srs-v2-20130331.xsd>.

Latest version available at: <http://www.upnp.org/schemas/av/srs.xsd>.

[27] – *XML Schema for ScheduledRecording LastChange Eventing*, UPnP Forum, September 30, 2008.

Available at: <http://www.upnp.org/schemas/av/srs-event-v1-20080930.xsd>.

Latest version available at: <http://www.upnp.org/schemas/av/srs-event.xsd>.

- [28] – *XML Schema for RenderingControl TransformSettings*, UPnP Forum, March 31, 2013.
Available at: <http://www.upnp.org/schemas/av/TransformSettings-v1-20130331.xsd>.
Latest version available at: <http://www.upnp.org/schemas/av/TransformSettings.xsd>.
- [29] – *XML Schema for ContentDirectory Metadata*, UPnP Forum, March 31, 2013.
Available at: <http://www.upnp.org/schemas/av/upnp-v4-20130331.xsd>.
Latest version available at: <http://www.upnp.org/schemas/av/upnp.xsd>.
- [30] – *The “xml:” Namespace*, November 3, 2004.
Available at: <http://www.w3.org/XML/1998/namespace>.
- [31] – *XML Schema for the “xml:” Namespace*.
Available at: <http://www.w3.org/2001/xml.xsd>.
- [32] – *Namespaces in XML*, Tim Bray, Dave Hollander, Andrew Layman, eds., W3C Recommendation, January 14, 1999.
Available at: <http://www.w3.org/TR/1999/REC-xml-names-19990114>.
- [33] – *XML Schema Part 1: Structures, Second Edition*, Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, W3C Recommendation, 28 October 2004.
Available at: <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028>.
- [34] – *XML Schema Part 2: Data Types, Second Edition*, Paul V. Biron, Ashok Malhotra, W3C Recommendation, 28 October 2004.
Available at: <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>.
- [35] – *XML Schema for XML Schema*.
Available at: <http://www.w3.org/2001/XMLSchema.xsd>.
- [36] – *Extensible Markup Language (XML) 1.0 (Third Edition)*, François Yergeau, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, eds., W3C Recommendation, February 4, 2004.
Available at: <http://www.w3.org/TR/2004/REC-xml-20040204>.
- [37] – *Unit conversion.org website, definitions of more than 2100 units in 78 categories*.
Available at: <http://www.unitconversion.org>.

3 Terms, definitions, symbols and abbreviations

For the purposes of this document, the terms and definitions given in [14] and the following subclauses 3.1 and 3.2 apply.

3.1 Provisioning terms

3.1.1 allowed

A

The definition or behavior is allowed.

3.1.2 conditionally allowed

CA

The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is allowed, otherwise it is not allowed.

3.1.3 conditionally required

CR

The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is required. Otherwise the definition or behavior is allowed as default unless specifically defined as not allowed.

3.1.4 required

R

The definition or behavior is required.

3.1.5

R/A

Used in a table column heading to indicate that each abbreviated entry in the column declares the provisioning status of the item named in the entry's row.

3.1.6

X

Vendor-defined, non-standard.

3.1.7

-D

Declares that the item referred to is deprecated, when it is appended to any of the other abbreviated provisioning terms.

3.1.8

CSV list (or CSV)

Comma separated value list. List—or one-dimensional array—of values contained in a string and separated by commas

3.2 Symbols

3.2.1

::

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.

4 Notations and Conventions

4.1 Notation

- UPnP interface names defined in the UPnP Device Architecture specification [14] are styled in **green bold underlined** text.
- UPnP interface names defined outside of the UPnP Device Architecture specification [14] are styled in **red italic underlined** text.
- Some additional non-interface names and terms are styled in *italic* text.
- Words that are emphasized are also styled in *italic* text. The difference between italic terms and italics for emphasis will be apparent by context.
- Strings that are to be taken literally are enclosed in “double quotes”.

4.1.1 Data Types

Data type definitions come from three sources:

- All state variable and action argument data types are defined in [14].
- Basic data types for properties are defined in [34].
- Additional data types for properties are defined in the XML schema(s) (see [3]) associated with this service.

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 shall 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 shall be accepted. Nevertheless, it is strongly recommended that all Boolean properties be represented as “*0*” and “*1*”.

4.1.2 Strings Embedded in Other Strings

Some string variables and arguments described in this document contain substrings that shall 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 subclause 4.2.2) 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
 - 3) represented as “\,” in individual substring entries in CSV lists
 - 4) 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

4.1.3 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 [19].

4.1.3.1 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 shall 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 in Table 1:

Table 1 — EBNF Operators

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

4.2 Derived Data Types

4.2.1 Summary

Subclause 4.2 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 in subclause 4.2.2 applies to both string data types.

4.2.2 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 [14], 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 (all values can be of different types). 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 ({a,b,c},{x, y, z}), where a, b, c, x, y and z are the types of the individual values in the subsequence and the subsequences 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 [14] (that is: allowed leading sign, allowed 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 [14]: **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 [34]: 0, false, 1, true.
- Escaping conventions for the comma and backslash characters are defined in 4.1.2.

- 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 2 — CSV Examples

Type refinement of string	Value	Comments
CSV (string) or CSV (xsd:string)	"+artist,-date"	List of 2 property sort criteria.
CSV (int) or CSV (xsd:integer)	"1,-5,006,0,+7"	List of 5 integers.
CSV (boolean) or CSV (xsd:Boolean)	"0,1,1,0"	List of 4 booleans
CSV (string) or CSV (xsd:string)	"Smith\, Fred,Jones\, Davey"	List of 2 names, "Smith, Fred" and "Jones, Davey"
CSV (i4 , string , ui2) 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 (i4) or CSV (xsd:int)	"3, 4"	Illegal CSV. White space is not allowed as part of an integer value.
CSV (string) 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 string , a department string and years-of-service ui2 or a name xsd:string, a department xsd:string and years-of-service xsd:unsignedShort.

4.3 Management of XML Namespaces in Standardized DCPs

UPnP specifications make extensive use of XML namespaces. This enables 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 shall 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 shall be locally unique to the document. Any valid XML no-colon-name may be used. And, in formal XML usage, different XML documents may use different namespace prefixes to refer to the same namespace. The creation and use of the namespace prefix was standardized by the W3C XML Committee in [32] 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, shall use the predefined namespace prefixes when referring to CDS properties (“upnp:”, “dc:”, etc).

All of the namespaces used in this specification are listed in Table 3 and Table 4. For each such namespace, Table 3 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 ScheduledRecording service depends on and refers to the ContentDirectory 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 4 to cross-reference additional namespace information. Table 4 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 2 for its associated schema.

The normative definitions for these namespaces are the documents referenced in Table 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.

Table 3 — Namespace Definitions

Standard Name-space Prefix	Namespace Name	Namespace Description	Normative Definition Document Reference
<i>AV Working Committee defined namespaces</i>			
atrs	urn:schemas-upnp-org:av:AllowedTransformSettings	AllowedTransformSettings and AllowedDefaultTransformSettings state variables for RenderingControl	[21]
av	urn:schemas-upnp-org:av:av	Common data types for use in AV schemas	[3]
avdt	urn:schemas-upnp-org:av:avdt	Datastructure Template	[2]
avs	urn:schemas-upnp-org:av:avs	Common structures for use in AV schemas	[4]
avt-event	urn:schemas-upnp-org:metadata-1-0/AVT/	Evented LastChange state variable for AVTransport	[5]
cds-event	urn:schemas-upnp-org:av:cds-event	Evented LastChange state variable for ContentDirectory	[7]
cm-dciu	urn:schemas-upnp-org:av:cm-deviceClockInfoUpdates	Evented DeviceClockInfoUpdates state variable for ConnectionManager	[9]
cm-ftrlst	urn:schemas-upnp-org:av:cm-featureList	FeatureList state variable for ConnectionManager	[9]

Standard Name-space Prefix	Namespace Name	Namespace Description	Normative Definition Document Reference
didl-lite	urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/	Structure and metadata for ContentDirectory	[7]
dmo	urn:schemas-upnp-org:av:dmo	Evented <u>DeviceMode</u> state variable for ContentDirectory	[7]
dmor	urn:schemas-upnp-org:av:dmor	<u>A_ARG_TYPE_DeviceModeRequest</u> state variable for ContentDirectory	[7]
dmos	urn:schemas-upnp-org:av:dmos	<u>DeviceModeStatus</u> state variable for ContentDirectory	[7]
pi	urn:schemas-upnp-org:av:pi	<u>PermissionsInfo</u> state variable for ContentDirectory	[7]
rcs-event	urn:schemas-upnp-org:metadata-1-0/RCS/	Evented <u>LastChange</u> state variable for RenderingControl	[21]
rii	urn:schemas-upnp-org:av:rii	<u>A_ARG_TYPE_RenderingInfoList</u> state variable for ConnectionManager	[9]
rpl	urn:schemas-upnp-org:av:rpl	<u>A_ARG_TYPE_PlaylistInfo</u> state variable for AVTransport	[5]
srs	urn:schemas-upnp-org:av:srs	Metadata and structure for ScheduledRecording	[25]
srs-event	urn:schemas-upnp-org:av:srs-event	Evented <u>LastChange</u> state variable for ScheduledRecording	[25]
trs	urn:schemas-upnp-org:av:TransformSettings	<u>TransformSettings</u> and <u>DefaultTransformSettings</u> state variables for RenderingControl	[21]
upnp	urn:schemas-upnp-org:metadata-1-0/upnp/	Metadata for ContentDirectory	[7]
<i>Externally defined namespaces</i>			
dc	http://purl.org/dc/elements/1.1/	Dublin Core	[13]
xsd	http://www.w3.org/2001/XMLSchema	XML Schema Language 1.0	[33], [34]
xsi	http://www.w3.org/2001/XMLSchema-instance	XML Schema Instance Document schema	[33] 2.6 & 3.2.7
xml	http://www.w3.org/XML/1998/namespace	The "xml:" Namespace	[30]

Table 4 — Schema-related Information

Standard Name-space Prefix	Relative URI and File Name ^a • Form 1, Form 2, Form3	Valid Root Element(s)	Schema Reference
<i>AV Working Committee Defined Namespaces</i>			
atrs	AllowedTransformSetting s-vn-yyyymmdd.xsd AllowedTransformSetting s-vn.xsd AllowedTransformSetting s.xsd	<TransformList>	[1]
av	av-vn-yyyymmdd.xsd av-vn.xsd av.xsd	n/a	[3]
avdt	avdt-vn-yyyymmdd.xsd avdt-vn.xsd avdt.xsd	<AVDT>	[2]
avs	avs-vn-yyyymmdd.xsd avs-vn.xsd avs.xsd	<Capabilities> <Features> <stateVariableValuePairs>	[4]
avt-event	avt-event-vn- yyyymmdd.xsd avt-event-vn.xsd avt-event.xsd	<Event>	[6]
cds-event	cds-event-vn- yyyymmdd.xsd cds-event-vn.xsd cds-event.xsd	<StateEvent>	[8]
cm-dciu	cm- deviceClockInfoUpdates- vn-yyyymmdd.xsd cm- deviceClockInfoUpdates -vn.xsd cm- deviceClockInfoUpdates. xsd	<DeviceClockInfoUpdates>	[10]
cm-ftrlst	cm-featureList-vn- yyyymmdd.xsd cm-featureList-vn.xsd cm-featureList.xsd	<Features>	[11]
didl-lite	didl-lite-vn- yyyymmdd.xsd didl-lite-vn.xsd didl-lite.xsd	<DIDL-Lite>	[15]
dmo	dmo-vn-yyyymmdd.xsd dmo-vn.xsd dmo.xsd	<DeviceMode>	[16]
dmor	dmor-vn-yyyymmdd.xsd dmor-vn.xsd dmor.xsd	<DeviceModeRequest>	[17]
dmos	dmos-vn-yyyymmdd.xsd dmos-vn.xsd dmos.xsd	<DeviceModeStatus>	[18]

Standard Name-space Prefix	Relative URI and File Name ^a • Form 1, Form 2, Form3	Valid Root Element(s)	Schema Reference
pi	pi-vn-yyyymmdd.xsd pi-vn.xsd pi.xsd	<PermissionsInfo>	[20]
rce-event	rce-event-vn-yyyymmdd.xsd rce-event-vn.xsd rce-event.xsd	<Event>	[22]
rii	rii-vn-yyyymmdd.xsd rii-vn.xsd rii.xsd	<rendererInfo>	[23]
rpl	rpl-vn-yyyymmdd.xsd rpl-vn.xsd rpl.xsd	<PlaylistInfo>	[24]
trs	TransformSettings-vn-yyyymmdd.xsd TransformSettings-vn.xsd TransformSettings.xsd	<TransformSettings>	[28]
srs	srs-vn-yyyymmdd.xsd srs-vn.xsd srs.xsd	<srs>	[26]
srs-event	srs-event-vn-yyyymmdd.xsd srs-event-vn.xsd srs-event.xsd	<StateEvent>	[27]
upnp	upnp-vn-yyyymmdd.xsd upnp-vn.xsd upnp.xsd	n/a	[29]
Externally Defined Namespaces			
dc	Absolute URL: http://dublincore.org/schemas/xmls/simpledc20021212.xsd		[12]
xsd	n/a	<schema>	[35]
xsi	n/a		n/a
xml	n/a		[31]
^a Absolute URIs are generated by prefixing the relative URIs with " http://www.upnp.org/schemas/av/ "			

4.3.1 Namespace Prefix Requirements

There are many occurrences in this specification of string data types that contain XML names (property names). These XML names in strings will not be processed under namespace-aware conditions. Therefore, all occurrences in instance documents of XML names in strings shall use the standard namespace prefixes as declared in Table 3. In order to properly process the XML documents described herein, control points and devices shall use namespace-aware XML processors [32] for both reading and writing. As allowed by [32], the namespace prefixes used in an instance document are at the sole discretion of the document creator. Therefore, the declared prefix for a namespace in a document may be different from the standard prefix. All devices shall be able to correctly process any valid XML instance document, even when it uses a non-standard prefix for ordinary XML names. However, it is strongly recommended that all devices use these standard prefixes for all instance documents to avoid confusion on the part of both human and machine readers. These standard prefixes are used in all descriptive text and all XML examples in this and related UPnP specifications. However, each individual specification may assume a default namespace for its descriptive text. In that case, names from that namespace may appear with no prefix.

The assumed default namespace, if any, for each UPnP AV specification is given in Table 5.

Note: all UPnP AV schemas declare attributes to be “unqualified”, so namespace prefixes are never used with AV Working Committee defined attribute names.

Table 5 — Default Namespaces for the AV Specifications

AV Specification Name	Default Namespace Prefix
AVTransport	avt-event
ConnectionManager	<i>n/a</i>
ContentDirectory	didl-lite
MediaRenderer	<i>n/a</i>
MediaServer	<i>n/a</i>
RenderingControl	rcs-event
ScheduledRecording	srs

4.3.2 Namespace Names, Namespace Versioning and Schema Versioning

The UPnP AV service specifications define several data structures (such as state variables and action arguments) whose format is an XML instance document that complies with one or more specific XML schemas, which define XML namespaces. Each namespace is uniquely identified by an assigned namespace name. The namespace names that are defined by the AV Working Committee are URNs. See Table 3 for a current list of namespace names. Additionally, each namespace corresponds to an XML schema document that provides a machine-readable representation of the associated namespace to enable automated validation of the XML (state variable or action parameter) instance documents.

Within an XML schema and XML instance document, the name of each corresponding namespace appears as the value of an `xmlns` attribute within the root element. Each `xmlns` attribute also includes a namespace prefix that is associated with that namespace in order to qualify and disambiguate element and attribute names that are defined within different namespaces. The schemas that correspond to the listed namespaces are identified by URI values that are listed in the `schemaLocation` attribute also within the root element (see subclause 4.3.3).

In order to enable both forward and backward compatibility, namespace names are permanently assigned and shall not change even when a new version of a specification changes the definition of a namespace. However, all changes to a namespace definition shall be backward-compatible. In other words, the updated definition of a namespace shall not invalidate any XML documents that comply with an earlier definition of that same namespace. This means, for example, that a namespace shall not be changed so that a new element or attribute becomes required in a conforming instance document. Although namespace names shall not change, namespaces still have version numbers that reflect a specific set of definitional changes. Each time the definition of a namespace is changed, the namespace's version number is incremented by one.

Whenever a new namespace version is created, a new XML schema document (.xsd) is created and published so that the new namespace definition is represented in a machine-readable form. Since a XML schema document is just a representation of a namespace definition, translation errors can occur. Therefore, it is sometime necessary to re-release a published schema in order to correct typos or other namespace representation errors. In order to easily identify the potential multiplicity of schema releases for the same namespace, the URI of each released schema shall conform to the following format (called Form 1):

Form 1: "http://www.upnp.org/schemas/av/" **schema-root-name** "-v" **ver** "-" **yyyymmdd** where

- **schema-root-name** is the name of the root element of the namespace that this schema represents.
- **ver** corresponds to the version number of the namespace that is represented by the schema.

- **yyyymmdd** is the year, month and day (in the Gregorian calendar) that this schema was released.

Table 4 identifies the URI formats for each of the namespaces that are currently defined by the UPnP AV Working Committee.

As an example, the original schema URI for the “rcs-event” namespace (that was released with the original publication of the UPnP AV service specifications in the year 2002) was “<http://www.upnp.org/schemas/av/rcs-event-v1-20020625.xsd>”. When the UPnP AV service specifications were subsequently updated in the year 2006, the URI for the updated version of the “rcs-event” namespace was “<http://www.upnp.org/schemas/av/rcs-event-v2-20060531.xsd>”. However, in 2006, the schema URI for the newly created “srs-event” namespace was “<http://www.upnp.org/schemas/av/srs-event-v1-20060531.xsd>”. Note the version field for the “srs-event” schema is “v1” since it was first version of that namespace whereas the version field for the “rcs-event” schema is “v2” since it was the second version of that namespace.

In addition to the dated schema URIs that are associated with each namespace, each namespace also has a set of undated schema URIs. These undated schema URIs have two distinct formats with slightly different meanings:

Form 2: “<http://www.upnp.org/schemas/av/>” *schema-root-name* “-v” **ver**
where **ver** is described above.

Form 3: “<http://www.upnp.org/schemas/av/>” *schema-root-name*

Form 2 of the undated schema URI is always linked to the most recent release of the schema that represents the version of the namespace indicated by **ver**. For example, the undated URI “.../av/rcs-event-v2.xsd” is linked to the most recent schema release of version 2 of the “rcs-event” namespace. Therefore, on May 31, 2006 (20060531), the undated schema URI was linked to the schema that is otherwise known as “.../av/rcs-event-v2-20060531.xsd”. Furthermore, if the schema for version 2 of the “rcs-event” namespace was ever re-released, for example to fix a typo in the 20060531 schema, then the same undated schema URI (“.../av/rcs-event-v2.xsd”) would automatically be updated to link to the updated version 2 schema for the “rcs-event” namespace.

Form 3 of the undated schema URI is always linked to the most recent release of the schema that represents the highest version of the namespace that has been published. For example, on June 25, 2002 (20020625), the undated schema URI “.../av/rcs-event.xsd” was linked to the schema that is otherwise known as “.../av/rcs-event-v1-20020625.xsd”. However, on May 31, 2006 (20060531), that same undated schema URI was linked to the schema that is otherwise known as “.../av/rcs-event-v2-20060531.xsd”.

When referencing a schema URI within an XML instance document or a referencing XML schema document, the following usage rules apply:

- All instance documents, whether generated by a service or a control point, shall use Form 3.
- All UPnP AV published schemas that reference other UPnP AV schemas shall also use Form 3.

Within an XML instance document, the definition for the `schemaLocation` attribute comes from the XML Schema namespace “<http://www.w3.org/2002/XMLSchema-instance>”. A single occurrence of the attribute can declare the location of one or more schemas. The `schemaLocation` attribute value consists of a whitespace separated list of values that is interpreted as a namespace name followed by its schema location URL. This pair-sequence is repeated as necessary for the schemas that need to be located for this instance document.

In addition to the schema URI naming and usage rules described above, each released schema shall contain a `version` attribute in the `<schema>` root element. Its value shall correspond to the format:

ver “-” **yyyymmdd** where **ver** and **yyyymmdd** are described above.

The `version` attribute provides self-identification of the namespace version and release date of the schema itself. For example, within the original schema released for the “rcs-event” namespace (`.../rcs-event-v2-20020625.xsd`), the `<schema>` root element contains the following attribute: `version="2-20020625"`.

4.3.3 Namespace Usage Examples

The `schemaLocation` attribute for XML instance documents comes from the XML Schema instance namespace “`http://www.w3.org/2002/XMLSchema-instance`”. A single occurrence of the attribute can declare the location of one or more schemas. The `schemaLocation` attribute value consists of a whitespace separated list of values: namespace name followed by its schema location URL. This pair-sequence is repeated as necessary for the schemas that need to be located for this instance document.

Example 1:

Sample *DIDL-Lite XML Instance Document*. Note that the references to the UPnP AV schemas do not contain any version or release date information. In other words, the references follow Form 3 from above. Consequently, this example is valid for all releases of the UPnP AV service specifications.

```
<?xml version="1.0" encoding="UTF-8"?>
<DIDL-Lite
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/"
  xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    urn:schemas-upnp-org:metadata-1-0/DIDL-Lite/
      http://www.upnp.org/schemas/av/didl-lite.xsd
    urn:schemas-upnp-org:metadata-1-0/upnp/
      http://www.upnp.org/schemas/av/upnp.xsd">
  <item id="18" parentID="13" restricted="0">
    ...
  </item>
</DIDL-Lite>
```

4.4 Vendor-defined Extensions

Whenever vendors create additional vendor-defined state variables, actions or properties, their assigned names and XML representation shall follow the naming conventions and XML rules as specified below in subclauses 4.4.1 to 4.4.4.

4.4.1 Vendor-defined Action Names

Vendor-defined action names shall begin with “**X**”. Additionally, it should be followed by an ICANN assigned domain name owned by the vendor followed by the underscore character (“_”). It shall then be followed by the vendor-assigned action name. The vendor-assigned action name shall not contain a hyphen character (“-”, 2D Hex in UTF-8) nor a hash character (“#”, 23 Hex in UTF-8). Vendor-assigned action names are case sensitive. The first character of the name shall be a US-ASCII letter (“A”-“Z”, “a”-“z”), US-ASCII digit (“0”-“9”), an underscore (“_”), or a non-experimental Unicode letter or digit greater than U+007F. Succeeding characters shall be a US-ASCII letter (“A”-“Z”, “a”-“z”), US-ASCII digit (“0”-“9”), an underscore (“_”), a period (“.”), a Unicode combiningchar, an extender, or a non-experimental Unicode letter or digit greater than U+007F. The first three letters shall not be “XML” in any combination of case.

4.4.2 Vendor-defined State Variable Names

Vendor-defined state variable names shall begin with “**X**”. Additionally, it should be followed by an ICANN assigned domain name owned by the vendor, followed by the underscore character (“_”). It shall then be followed by the vendor-assigned state variable name. The vendor-assigned state variable name shall not contain a hyphen character (“-”, 2D Hex in UTF-8). Vendor-assigned action names are case sensitive. The first character of the name shall be a US-ASCII letter (“A”-“Z”, “a”-“z”), US-ASCII digit (“0”-“9”), an underscore (“_”), or a non-experimental Unicode letter or digit greater than U+007F. Succeeding characters shall be a US-ASCII letter (“A”-“Z”, “a”-“z”), US-ASCII digit (“0”-“9”), an underscore (“_”), a period (“.”), a Unicode combiningchar, an extender, or a non-experimental Unicode letter or digit greater than U+007F.

a US-ASCII letter ("A"-"Z", "a"-"z"), US-ASCII digit ("0"-"9"), an underscore ("_"), a period ("."), a Unicode combiningchar, an extender, or a non-experimental Unicode letter or digit greater than U+007F. The first three letters shall not be "XML" in any combination of case.

4.4.3 Vendor-defined XML Elements and attributes

UPnP vendors may add non-standard elements and attributes to a UPnP standard XML document, such as a device or service description. Each addition shall be scoped by a vendor-owned XML namespace. Arbitrary XML shall be enclosed in an element that begins with "X," and this element shall be a sub element of a standard complex type. Non-standard attributes may be added to standard elements provided these attributes are scoped by a vendor-owned XML namespace and begin with "X".

4.4.4 Vendor-defined Property Names

UPnP vendors may add non-standard properties to the ContentDirectory service. Each property addition shall be scoped by a vendor-owned namespace. The vendor-assigned property name shall not contain a hyphen character ("-", 2D Hex in UTF-8). Vendor-assigned property names are case sensitive. The first character of the name shall be a US-ASCII letter ("A"-"Z", "a"-"z"), US-ASCII digit ("0"-"9"), an underscore ("_"), or a non-experimental Unicode letter or digit greater than U+007F. Succeeding characters shall be a US-ASCII letter ("A"-"Z", "a"-"z"), US-ASCII digit ("0"-"9"), an underscore ("_"), a period ("."), a Unicode combiningchar, an extender, or a non-experimental Unicode letter or digit greater than U+007F. The first three letters shall not be "XML" in any combination of case.

5 Service Modeling Definitions

5.1 Service Type

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

urn:schemas-upnp-org:service:*RenderingControl*:3

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

5.2 State Variables

5.2.1 State Variable Overview

Table 6 — State Variables

Variable Name	R/A a	Data Type	Allowed Value	Default Value	Eng. Units
<u>LastChange</u>	<u>R</u>	<u>string</u>	See 5.2.2		
<u>PresetNameList</u>	<u>R</u>	<u>string</u>	CSV b (<u>string</u>) See 5.2.3		
<u>Brightness</u>	<u>CR</u> c	<u>ui2</u>	See Table 7 and 5.2.4		
<u>Contrast</u>	<u>CR</u> c	<u>ui2</u>	See Table 8 and 5.2.5		
<u>Sharpness</u>	<u>CR</u> c	<u>ui2</u>	See Table 9 and 5.2.6		
<u>RedVideoGain</u>	<u>CR</u> c	<u>ui2</u>	See Table 10 and 5.2.7		
<u>GreenVideoGain</u>	<u>CR</u> c	<u>ui2</u>	See Table 11 and 5.2.8		
<u>BlueVideoGain</u>	<u>CR</u> c	<u>ui2</u>	See Table 12 and 5.2.9		
<u>RedVideoBlackLevel</u>	<u>CR</u> c	<u>ui2</u>	See Table 13 and 5.2.10		
<u>GreenVideoBlackLevel</u>	<u>CR</u> c	<u>ui2</u>	See Table 14 and 5.2.11		
<u>BlueVideoBlackLevel</u>	<u>CR</u> c	<u>ui2</u>	See Table 15 and 5.2.12		
<u>ColorTemperature</u>	<u>CR</u> c	<u>ui2</u>	See Table 16 and 5.2.13		
<u>HorizontalKeystone</u>	<u>CR</u> c	<u>i2</u>	See Table 17 and 5.2.14		
<u>VerticalKeystone</u>	<u>CR</u> c	<u>i2</u>	See Table 18 and 5.2.15		
<u>Mute</u>	<u>CR</u> c	<u>boolean</u>	See 5.2.16		
<u>Volume</u> d	<u>CR</u> c	<u>ui2</u>	See Table 19 and 5.2.17		
<u>VolumeDB</u>	<u>CR</u> c	<u>i2</u>	See Table 20 and 5.2.18		1/256 dB
<u>Loudness</u>	<u>CR</u> c	<u>boolean</u>	See 5.2.19		
<u>AllowedTransformSettings</u>	<u>CR</u> c	<u>string</u>	See 5.2.20		
<u>TransformSettings</u>	<u>CR</u> c	<u>string</u>	See 5.2.21		
<u>AllowedDefaultTransformSettings</u>	<u>CR</u> c	<u>string</u>	See 5.2.22		
<u>DefaultTransformSettings</u>	<u>CR</u> c	<u>string</u>	See 5.2.23		
<u>A_ARG_TYPE_Channel</u>	<u>R</u>	<u>string</u>	See Table 21 and 5.2.24		
<u>A_ARG_TYPE_InstanceID</u>	<u>R</u>	<u>ui4</u>	See 5.2.25		
<u>A_ARG_TYPE_PresetName</u>	<u>R</u>	<u>string</u>	See Table 22 and 5.2.26		
<u>A_ARG_TYPE_DeviceUDN</u>	<u>CR</u> c	<u>string</u>	See 5.2.27		
<u>A_ARG_TYPE_ServiceType</u>	<u>CR</u> c	<u>string</u>	See 5.2.28		
<u>A_ARG_TYPE_ServiceID</u>	<u>CR</u> c	<u>string</u>	See 5.2.29		
<u>A_ARG_TYPE_StateVariableValuePairs</u>	<u>CR</u> c	<u>string</u>	See 5.2.30		
<u>A_ARG_TYPE_StateVariableList</u>	<u>CR</u> c	<u>string</u>	CSV (<u>string</u>) See 5.2.31		
<i>Non-standard state variables implemented by an UPnP vendor go here.</i>	<u>X</u>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>

Variable Name	R/A a	Data Type	Allowed Value	Default Value	Eng. Units
a <u>R</u> = required, <u>A</u> = allowed, <u>X</u> = Non-standard. b CSV stands for Comma-Separated Value list. The type between brackets denotes the UPnP data type used for the elements inside the list. CSV is defined more formally in the ContentDirectory service template. c See subclause of the relevant state variable for conditions under which implementation of this state variable is required. d The <u>Volume</u> and <u>VolumeDB</u> state variables are defined as a pair. Therefore, each implementation of this service shall either support both of them or support none of them. At all times, these two state variables shall remain synchronized with each other (that is: both of them shall always represent the same volume setting).					

Table 7 — allowedValueRange for Brightness

	Value	R/A
<u>minimum</u>	<u>0</u>	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>step</u>	<u>1</u>	<u>R</u>

Table 8 — allowedValueRange for Contrast

	Value	R/A
<u>minimum</u>	<u>0</u>	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>step</u>	<u>1</u>	<u>R</u>

Table 9 — allowedValueRange for Sharpness

	Value	R/A
<u>minimum</u>	<u>0</u>	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>step</u>	<u>1</u>	<u>R</u>

Table 10 — allowedValueRange for RedVideoGain

	Value	R/A
<u>minimum</u>	<u>0</u>	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>step</u>	<u>1</u>	<u>R</u>

Table 11 — allowedValueRange for GreenVideoGain

	Value	R/A
<u>minimum</u>	<u>0</u>	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>step</u>	<u>1</u>	<u>R</u>

Table 12 — allowedValueRange for BlueVideoGain

	Value	R/A
<u>minimum</u>	<u>0</u>	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>step</u>	<u>1</u>	<u>R</u>

Table 13 — allowedValueRange for RedVideoBlackLevel

	Value	R/A
<u>minimum</u>	<u>0</u>	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>step</u>	<u>1</u>	<u>R</u>

Table 14 — allowedValueRange for GreenVideoBlackLevel

	Value	R/A
<u>minimum</u>	<u>0</u>	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>step</u>	<u>1</u>	<u>R</u>

Table 15 — allowedValueRange for BlueVideoBlackLevel

	Value	R/A
<u>minimum</u>	<u>0</u>	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>step</u>	<u>1</u>	<u>R</u>

Table 16 — allowedValueRange for ColorTemperature

	Value	R/A
<u>minimum</u>	<u>0</u>	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>step</u>	<u>1</u>	<u>R</u>

Table 17 — allowedValueRange for HorizontalKeystone

	Value	R/A
<u>minimum</u>	<i>Vendor defined</i> (shall be ≤ 0)	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>Step</u>	<u>1</u>	<u>R</u>

Table 18 — allowedValueRange for VerticalKeystone

	Value	R/A
<u>minimum</u>	<i>Vendor defined</i> (shall be ≤ 0)	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>step</u>	<u>1</u>	<u>R</u>

Table 19 — allowedValueRange for Volume

	Value	R/A
<u>minimum</u>	0	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>Step</u>	<u>1</u>	<u>R</u>

Table 20 — allowedValueRange for VolumeDB

	Value	R/A
<u>minimum</u>	<i>Vendor defined</i>	<u>R</u>
<u>maximum</u>	<i>Vendor defined</i>	<u>R</u>
<u>step</u>	<i>Vendor defined</i>	<u>A</u>

Table 21 — allowedValueList for **A_ARG_TYPE_Channel**

Value	R/A	Description
<u>"Master"</u>	<u>R</u>	Master volume
<u>"LF"</u>	<u>A</u>	Left Front
<u>"RF"</u>	<u>A</u>	Right Front
<u>"CF"</u>	<u>A</u>	Center Front
<u>"LFE"</u>	<u>A</u>	Low Frequency Enhancement
<u>"LS"</u>	<u>A</u>	Left Surround
<u>"RS"</u>	<u>A</u>	Right Surround
<u>"LFC"</u>	<u>A</u>	Left of Center [in front]
<u>"RFC"</u>	<u>A</u>	Right of Center [in front]
<u>"SD"</u>	<u>A</u>	Surround [rear]
<u>"SL"</u>	<u>A</u>	Side Left [left wall]
<u>"SR"</u>	<u>A</u>	Side Right [right wall]
<u>"T"</u>	<u>A</u>	Top [overhead]
<u>"B"</u>	<u>A</u>	Bottom
<u>"BC"</u>	<u>A</u>	Back Center
<u>"BL"</u>	<u>A</u>	Back Left
<u>"BR"</u>	<u>A</u>	Back Right
<u>Vendor-defined</u>	<u>X</u>	

Table 22 — allowedValueList for **A_ARG_TYPE_PresetName**

Value	R/A
<u>"FactoryDefaults"</u>	<u>R</u>
<u>"InstallationDefaults"</u>	<u>A</u>
<u>Vendor defined</u>	<u>X</u>

5.2.2 **LastChange**

This required state 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 subclause 5.3.2 for additional information.

The format of this state variable conforms to the XML schema described in [22]. The following *rsc-event XML document* illustrates a typical example of the schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<Event
  xmlns="urn:schemas-upnp-org:metadata-1-0/RCS/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    urn:schemas-upnp-org:metadata-1-0/RCS/
    http://www.upnp.org/schemas/av/rsc-event-v1-20060531.xsd">
  <InstanceID val="0">
    <Brightness val="36"/>
    <Contrast val="54"/>
    ...
  </InstanceID>
  <InstanceID val="1">
    <Mute channel="Master" val="0"/>
    <Volume channel="CF" val="24"/>
    ...
  </InstanceID>
```

```

<InstanceID val="2">
  <AllowedTransformSettings val="
    <?xml version="1.0" encoding="UTF-8"?>
    <TransformList
      xmlns="urn:schemas-upnp-org:av:AllowedTransformSettings"
      xmlns:xsi"http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation=
        "urn:schemas-upnp-org:av:AllowedTransformSettings
http://www.upnp.org/schemas/av/AllowedTransformSettings.xsd">
      <transform name="Rotation">
        <allowedValueRange inactiveValue="0">
          <minimum>0</minimum>
          <maximum>270</maximum>
          <step>90</step>
        </allowedValueRange>
      </transform>
      <transform name="Zoom">
        <allowedValueRange unit="pct" info="Zoom factor in percent">
          <minimum>50</minimum>
          <maximum>400</maximum>
          <step>1</step>
        </allowedValueRange>
        <allowedValueList info="Pre-defined Zoom values"
inactiveValue="None">
          <allowedValue>None</allowedValue>
          <allowedValue>N/A</allowedValue>
          <allowedValue>Zoom 1</allowedValue>
          <allowedValue>Zoom 2</allowedValue>
        </allowedValueList>
      </transform>
      <!-- ... remaining transforms allowed for this instance -->
    </TransformList>"
  />
  <TransformSettings val="
    <?xml version="1.0" encoding="UTF-8"?>
    <TransformSettings xmlns=
      "urn:schemas-upnp-org:av:TransformSettings"
      xmlns:xsi"http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:schemas-upnp-org:av:TransformSettings
http://www.upnp.org/schemas/av/TransformSettings.xsd">
      <transform name="Rotation">
        <value>0</value>
      </transform>
      <transform name="Zoom">
        <value>100</value>
        <value index="1">None</value>
      </transform>
      <!-- ... remaining transform settings for this instance -->
    </TransformSettings>"
  />
  ...
</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 RenderingControl 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, respectively. Additionally, the Mute setting on the Master channel of instance 1 has been set to 0 (false) (that is: muting has been turned off) and the Volume of the Center Front channel of instance 1 has been set to 24. For instance 2 there are 2 transforms, Rotation and Zoom. The Rotation transform has the set of allowed values of [0, 90, 180, 270, 360, 450, 540, 630, 720, 810, 900, 990].

currently the value of 0 (no rotation). The [Zoom](#) transform has the allowed value range of 50% to 400% and has been set to 100%. Additionally, the [Zoom](#) transform has a number of pre-defined vendor-specific allowed values. For a list of pre-defined transforms see Annex B.

Note: Only the audio-related state variables 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 for the affected state variable will appear within the [InstanceID](#) element. The previous state variable element for the affected state variable shall 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 they may appear in any order.

For example, when the [Brightness](#) of instance 0 changes from 26 to 54 then to 48, the [Brightness](#) of instance 1 changes from 54 to 35, then to 11, and a transform has been performed on instance 2 such that the [TransformSettings](#) are modified, [LastChange](#) is set to the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<Event
  xmlns="urn:schemas-upnp-org:metadata-1-0/RCS/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    urn:schemas-upnp-org:metadata-1-0/RCS/
    http://www.upnp.org/schemas/av/rcs-event-v1-20060531.xsd">
  <InstanceID val="0">
    <Brightness val="48"/>
  </InstanceID>
  <InstanceID val="1">
    <Brightness val="11"/>
  </InstanceID>
  <InstanceID val="2">
    <TransformSettings val="
      <!-- ... the following text needs to be XML escaped -->
      <?xml version="1.0" encoding="UTF-8"?>
      <TransformSettings xmlns=
        "urn:schemas-upnp-org:av:TransformSettings"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:schemas-upnp-org:av:TransformSettings
        http://www.upnp.org/schemas/av/TransformSettings.xsd">
        <transform name="Rotation">
          <value>90</value>
        </transform>
        <transform name="Zoom">
          <value>100</value>
        </transform>
      </TransformSettings>"
    />
  </InstanceID>
</Event>
```

The [LastChange](#) state variable is evented using the standard UPnP event mechanism. All other state variables, except the state variables [AllowedDefaultTransformSettings](#) and [DefaultTransformSettings](#), are indirectly evented via the [LastChange](#) state variable event. When a new instance is created by invoking the [ConnectionManager::PrepareForConnection\(\)](#) action, the [LastChange](#) state variable shall incorporate the new instance and thus be evented with the initial values. Refer to subclause 5.3.2 for additional details.

Note that the [LastChange](#) state variable is XML and therefore needs to be escaped using the normal XML rules (see [36] – 2.4 Character Data) before being transmitted as an event.

Note that for all values of state variables that contain XML, the content shall be escaped before inserting it inside LastChange.

Note when a new instance is created, the initial values for transforms should be equal to the values in the DefaultTransformSettings state variable, or the transform's inactive value when no default value is established.

5.2.3 PresetNameList

This required state variable contains a comma-separated list 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.

5.2.4 Brightness

This conditionally required state variable shall be implemented if the GetBrightness() or SetBrightness() actions are implemented. The unsigned integer state 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.

5.2.5 Contrast

This conditionally required state variable shall be implemented if the GetContrast() or SetContrast() actions are implemented. The unsigned integer state 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.

5.2.6 Sharpness

This conditionally required state variable shall be implemented if the GetSharpness() or SetSharpness() actions are implemented. The unsigned integer state 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.

5.2.7 RedVideoGain

This conditionally required state variable shall be implemented if the GetRedVideoGain() or SetRedVideoGain() actions are implemented. The unsigned integer state 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 subclause 5.2.32.1 for additional information.

5.2.8 GreenVideoGain

This conditionally required state variable shall be implemented if the GetGreenVideoGain() or SetGreenVideoGain() actions are implemented. The unsigned integer state 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 subclause 5.2.32.1 for additional information.

5.2.9 BlueVideoGain

This conditionally required state variable shall be implemented if the [GetBlueVideoGain\(\)](#) or [SetBlueVideoGain\(\)](#) actions are implemented. The unsigned integer state 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 subclause 5.2.32.1 for additional information.

5.2.10 RedVideoBlackLevel

This conditionally required state variable shall be implemented if the [GetRedVideoBlackLevel\(\)](#) or [SetRedVideoBlackLevel\(\)](#) actions are implemented. The unsigned integer state 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 subclause 5.2.32.1 for additional information.

5.2.11 GreenVideoBlackLevel

This conditionally required state variable shall be implemented if the [GetGreenVideoBlackLevel\(\)](#) or [SetGreenVideoBlackLevel\(\)](#) actions are implemented. The unsigned integer state 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 subclause 5.2.32.1 for additional information.

5.2.12 BlueVideoBlackLevel

This conditionally required state variable shall be implemented if the [GetBlueVideoBlackLevel\(\)](#) or [SetBlueVideoBlackLevel\(\)](#) actions are implemented. The unsigned integer state 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 subclause 5.2.32.1 for additional information.

5.2.13 ColorTemperature

This conditionally required state variable shall be implemented if the [GetColorTemperature\(\)](#) or [SetColorTemperature\(\)](#) actions are implemented. The unsigned integer state 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 (that is: biased towards yellow/orange) and higher values produce cooler shades of white (that is: biased towards blue).

5.2.14 HorizontalKeystone

This conditionally required state variable shall be implemented if the [GetHorizontalKeystone\(\)](#) or [SetHorizontalKeystone\(\)](#) actions are implemented. The signed integer state 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