INTERNATIONAL STANDARD

ISO/IEC 29341-20-14

First edition
2017-09

Information technology — UPnP Device Architecture —

Part 20-14: Audio video device control protocol — Level 4 — Scheduled recording service

Technologies de l'information — Architecture de dispositif UPnP —

Partie 20-14: Protocole de contrôle de dispositif audio-vidéo — Niveau 4 — Service d'enregistrement programmé

Reference number
ISO/IEC 29341-20-14:2017(E)

© 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**
2 Normative references **2**
3 Terms, definitions, symbols and abbreviations **5**
3.1 Provisioning terms 5
3.2 Terms specific to ScheduledRecording service 5
3.2.1 CDS 5
3.2.2 EPG 6
3.2.3 SRS 6
3.2.4 CDS object 6
3.2.5 User Channel 6
3.2.6 Channel Group 6
3.2.7 Channel Line-up 6
3.2.8 object 6
3.2.9 class 6
3.2.10 object Modification 6
3.2.11 *recordSchedule* 6
3.2.12 Conflicting *recordSchedule* 6
3.2.13 *recordTask* 7
3.2.14 Conflicting *recordTask* 7
3.2.15 *recordScheduleParts* 7
3.2.16 Property-set Data Types 7
3.2.17 Property 7
3.2.18 Member Property 8
3.2.19 Supported Member Property 8
3.2.20 Multi-valued property 8
3.2.21 Single-valued property 8
3.2.22 XML Document 8
3.2.23 XML Fragment 9
3.2.24 actualScheduledStartDateTime 9
3.2.25 actualStartDateTime 9
3.2.26 actualScheduledEndDateTime 9
3.2.27 actualEndDateTime 9
3.2.28 actualScheduledDuration 9
3.2.29 Lexical Sort Order 10
3.2.30 Lexical Matching 10
3.2.31 Simple Non-case-sensitive Sort Order 10
3.2.32 Simple Non-case-sensitive Matching 11
3.2.33 Numeric Sort Order 11
3.2.34 Boolean Sort Order 11
3.2.35 Sequenced Sort 11
3.2.36 Sequenced Lexical Sort 11
3.2.37 Sequenced Numeric Sort 11
3.2.38 Lexical Numeric Sort 11
3.2.39 *type* Relationship Sort 12
3.3 Symbols 12
4 Notations and Conventions **12**

© ISO/IEC 2017 – All rights reserved

4.1 Notation 12
4.1.1 Data Types 12
4.1.2 Strings Embedded in Other Strings 12
4.1.3 Extended Backus-Naur Form 13
4.2 Derived Data Types 13
4.2.1 Summary 13
4.2.2 CSV Lists 14
4.3 Management of XML Namespaces in Standardized DCPs 15
4.3.1 Namespace Prefix Requirements 19
4.3.2 Namespace Names, Namespace Versioning and Schema Versioning 20
4.3.3 Namespace Usage Examples 22
4.4 Vendor-defined Extensions 22
4.4.1 Vendor-defined Action Names 22
4.4.2 Vendor-defined State Variable Names 23
4.4.3 Vendor-defined XML Elements and attributes 23
4.4.4 Vendor-defined Property Names 23
5 Service Modeling Definitions 23
5.1 ServiceType 23
5.2 ScheduledRecording Service Architecture 23
5.2.1 *recordSchedule* 23
5.2.2 *recordTask* 25
5.3 State Variables 26
5.3.1 State Variable Overview 26
5.3.2 *SortCapabilities* 28
5.3.3 *SortLevelCapability* 28
5.3.4 *StateUpdateID* 28
5.3.5 *LastChange* 29
5.3.6 *A_ARG_TYPE_PropertyList* 32
5.3.7 *A_ARG_TYPE_DataTypeID* 32
5.3.8 *A_ARG_TYPE_ObjectID* 32
5.3.9 *A_ARG_TYPE_ObjectIDList* 32
5.3.10 *A_ARG_TYPE_PropertyInfo* 33
5.3.11 *A_ARG_TYPE_Index* 33
5.3.12 *A_ARG_TYPE_Count* 33
5.3.13 *A_ARG_TYPE_SortCriteria* 33
5.3.14 *A_ARG_TYPE_RecordSchedule* 33
5.3.15 *A_ARG_TYPE_RecordTask* 34
5.3.16 *A_ARG_TYPE_RecordScheduleParts* 34
5.4 Eventing and Moderation 35
5.5 Actions 35
5.5.1 Action Overview 35
5.5.2 *GetSortCapabilities()* 36
5.5.3 *GetPropertyList()* 37
5.5.4 *GetAllowedValues()* 38
5.5.5 *GetStateUpdateID()* 40
5.5.6 *BrowseRecordSchedules()* 40
5.5.7 *BrowseRecordTasks()* 45
5.5.8 *CreateRecordSchedule()* 47

 © ISO/IEC 2017 – All rights reserved

5.5.9 *DeleteRecordSchedule()* 49
5.5.10 *GetRecordSchedule()* 50
5.5.11 *EnableRecordSchedule()* 51
5.5.12 *DisableRecordSchedule()* 52
5.5.13 *DeleteRecordTask()* 54
5.5.14 *GetRecordTask()* 54
5.5.15 *EnableRecordTask()* 55
5.5.16 *DisableRecordTask()* 57
5.5.17 *ResetRecordTask()* 58
5.5.18 *GetRecordScheduleConflicts()* 59
5.5.19 *GetRecordTaskConflicts()* 60
5.5.20 Common Error Codes 61
5.6 State Diagram of *recordTask* 62
5.6.1 A Full-Featured State Diagram 62
5.6.2 A Minimal-Implementation State Diagram 68
5.6.3 *recordTask* State Example 70
5.7 ScheduledRecording Service Priority Model 71
5.7.1 Introduction of the ScheduledRecording Service Priority Model 71
5.7.2 Ordered Priority within Each Priority Level 72
5.7.3 Setting the Initial Priority Level of a *recordSchedule* 73
5.7.4 Sorting *recordSchedule* Instances Based on their Current Priority Settings 75
5.8 Theory of Operation 76
5.8.1 Introduction 76
5.8.2 Checking the Capabilities of a ScheduledRecording Service 76
5.8.3 Adding a Scheduled Recording Entry to the List 88
5.8.4 Deleting a *recordSchedule* 106
5.8.5 Browsing *recordSchedule* and *recordTask* instances 106
5.8.6 Rating System 113
5.8.7 Conflict Detection and Resolution 113
6 XML Service Description 115
7 Test 125
Annex A (normative) srs XML Document 126
A.1 *A_ARG_TYPE_RecordSchedule AVDT XML Document* 126
A.2 *A_ARG_TYPE_RecordTask AVDT XML Document* 127
A.3 *A_ARG_TYPE_RecordScheduleParts AVDT XML Document* 127
Annex B (normative) AV Working Committee Extended Properties 129
B.1 Base Properties 129
B.1.1 *@id* 129
B.1.2 *title* 130
B.1.3 *class* 130
B.1.4 *additionalStatusInfo* 130
B.1.5 *cdsReference* 131
B.2 Priority Properties 132
B.2.1 *priority* 132
B.2.2 *desiredPriority* 133
B.3 Output Control Properties 135
B.3.1 *recordDestination* 135

© ISO/IEC 2017 – All rights reserved

B.3.2 *desiredRecordQuality* 137
B.4 Content Identification Related Properties 140
B.4.1 *scheduledCDSObjectID* 140
B.4.2 *scheduledChannelID* 141
B.4.3 *scheduledStartDateTime* 143
B.4.4 *scheduledDuration* 144
B.4.5 *scheduledProgramCode* 144
B.5 Matching Content Criteria Properties 145
B.5.1 *matchingName* 145
B.5.2 *matchingID* 146
B.6 Matching Qualifying Criteria Properties 147
B.6.1 *matchingChannelID* 148
B.6.2 *matchingStartDateTimeRange* 149
B.6.3 *matchingDurationRange* 149
B.6.4 *matchingRatingLimit* 150
B.6.5 *matchingEpisodeType* 152
B.7 Content Control Properties 153
B.7.1 *totalDesiredRecordTasks* 153
B.7.2 *scheduledStartDateTimeAdjust* 154
B.7.3 *scheduledDurationAdjust* 154
B.7.4 *activePeriod* 155
B.7.5 *durationLimit* 155
B.7.6 *channelMigration* 156
B.7.7 *timeMigration* 157
B.7.8 *allowDuplicates* 157
B.8 Storage Related Properties 157
B.8.1 *persistedRecordings* 158
B.9 Schedule State Properties 159
B.9.1 *scheduleState* 159
B.9.2 *abnormalTasksExist* 161
B.10 Statistics Properties 161
B.10.1 *currentRecordTaskCount* 161
B.10.2 *totalCreatedRecordTasks* 161
B.10.3 *totalCompletedRecordTasks* 162
B.11 Task General Properties 162
B.11.1 *recordScheduleID* 162
B.11.2 *recordedCDSObjectID* 162
B.12 Task Content Identification Properties 163
B.12.1 *taskCDSObjectID* 164
B.12.2 *taskChannelID* 164
B.12.3 *taskStartDateTime* 165
B.12.4 *taskDuration* 166
B.12.5 *taskProgramCode* 166
B.12.6 *recordQuality* 166
B.13 Task Matched Content Criteria Properties 169
B.13.1 *matchedName* 169
B.13.2 *matchedID* 169
B.14 Task Matched Qualifying Criteria Properties 170
B.14.1 *matchedRating* 170

 © ISO/IEC 2017 – All rights reserved

B.14.1.1 *matchedRating@type* 170
B.14.1.2 *matchedRating@equivalentAge* 171
B.14.2 *matchedEpisodeType* 171
B.15 Task Matched Content Control Properties 171
B.15.1 *taskStartDateTimeAdjust* 171
B.15.2 *taskDurationAdjust* 172
B.15.3 *taskDurationLimit* 172
B.15.3.1 *taskDurationLimit@effect* 172
B.15.4 *taskChannelMigration* 173
B.15.5 *taskTimeMigration* 173
B.16 Task State Properties 173
B.16.1 *taskState* 174
B.17 ContentDirectory Service Imported Properties 182
Annex C (normative) AV Working Committee Class Definitions 186
C.1 Class Hierarchy 186
C.1.1 Relationships between Classes and Properties 188
C.1.2 *recordScheduleParts* Properties 189
C.1.3 *recordSchedule* Properties 193
C.1.4 *recordTask* Properties 197
C.2 Class Definitions 199
C.3 *object* Base Class 200
C.3.1 *object.recordSchedule* Class 201
C.3.2 *object.recordTask* Class 211
Annex D (normative) EBNF Syntax Definitions 213
D.1 Priority Syntax 213
D.2 Date&time Syntax 213
D.3 Class Name Syntax 214
Annex E (informative) ScheduledRecording Service Relationship to ContentDirectory Service 215
Annex F (informative) ScheduledRecording Service Relationship to EPG 216
Annex G (informative) AVDT Examples 217
G.1 *A_ARG_TYPE_RecordSchedule* AVDT Example 217
G.2 *A_ARG_TYPE_RecordTask* AVDT Example 234
G.3 *A_ARG_TYPE_RecordScheduleParts* AVDT Example 253
Annex H (informative) Bibliography 269

© ISO/IEC 2017 – All rights reserved

List of Tables

Table 1 — Properties in XML 8
Table 2 — EBNF Operators 13
Table 3 — CSV Examples 15
Table 4 — Namespace Definitions 16
Table 5 — Schema-related Information 18
Table 6 — Default Namespaces for the AV Specifications 20
Table 7 — State Variables 26
Table 8 — Allowed values for the *DataTypeID* argument 28
Table 9 — Allowed Elements in <StateEvent> Element 29
Table 10 — Eventing and Moderation 35
Table 11 — Actions 35
Table 12 — Arguments for *GetSortCapabilities()* 36
Table 13 — Error Codes for *GetSortCapabilities()* 37
Table 14 — Arguments for *GetPropertyList()* 37
Table 15 — Error Codes for *GetPropertyList()* 38
Table 16 — Arguments for *GetAllowedValues()* 38
Table 17 — Error Codes for *GetAllowedValues()* 39
Table 18 — Arguments for *GetStateUpdateID()* 40
Table 19 — Error Codes for *GetStateUpdateID()* 40
Table 20 — Arguments for *BrowseRecordSchedules()* 40
Table 21 — Error Codes for *BrowseRecordSchedules()* 45
Table 22 — Arguments for *BrowseRecordTasks()* 45
Table 23 — Error Codes for *BrowseRecordTasks()* 47
Table 24 — Arguments for *CreateRecordSchedule()* 47
Table 25 — Error Codes for *CreateRecordSchedule()* 49
Table 26 — Arguments for *DeleteRecordSchedule()* 50
Table 27 — Error Codes for *DeleteRecordSchedule()* 50
Table 28 — Arguments for *GetRecordSchedule()* 50
Table 29 — Error Codes for *GetRecordSchedule()* 51
Table 30 — Arguments for *EnableRecordSchedule()* 52
Table 31 — Error Codes for *EnableRecordSchedule()* 52
Table 32 — Arguments for *DisableRecordSchedule()* 53
Table 33 — Error Codes for *DisableRecordSchedule()* 53
Table 34 — Arguments for *DeleteRecordTask()* 54
Table 35 — Error Codes for *DeleteRecordTask()* 54
Table 36 — Arguments for *GetRecordTask()* 55
Table 37 — Error Codes for *GetRecordTask()* 55
Table 38 — Arguments for *EnableRecordTask()* 56
Table 39 — Error Codes for *EnableRecordTask()* 56
Table 40 — Arguments for *DisableRecordTask()* 57
Table 41 — Error Codes for *DisableRecordTask()* 58

 © ISO/IEC 2017 – All rights reserved

Table 42 — Arguments for *ResetRecordTask()* 58
Table 43 — Error Codes for *ResetRecordTask()* 59
Table 44 — Arguments for *GetRecordScheduleConflicts()* 59
Table 45 — Error Codes for *GetRecordScheduleConflicts()* 60
Table 46 — Arguments for *GetRecordTaskConflicts()* 60
Table 47 — Error Codes for *GetRecordTaskConflicts()* 61
Table 48 — Common Error Codes 61
Table 49 — *recordTask* State Timeline 71
Table 50 — Example 1: Fewer *recordSchedule* instances than the Number of Supported Priority Levels. 72
Table 51 — Example 2: More *recordSchedule* instances than the Number of Supported Priority Levels 73
Table 52 — Existing *recordSchedule* Priorities 74
Table 53 — *desiredPriority* Property Set to "*RS-C*" 74
Table 54 — *desiredPriority* Property Set to "*HIGHEST*", "*L1_HI*", or "*RS-A*" 75
Table 55 — *desiredPriority* Property Set to "*LOWEST*", "*L3_LOW*", or "*RS-B*" 75
Table 56 — *desiredPriority* Property Set to "*RS-C*" 75
Table B.1 — Base Properties Overview 129
Table B.2 — Allowed values for the *class* Property 130
Table B.3 — Priority Properties 132
Table B.4 — Allowed values for the *priority* Property 132
Table B.5 — Primary allowed values for the *desiredPriority* Property 133
Table B.6 — Additional allowed values for the *desiredPriority* Property 134
Table B.7 — Allowed values for the *desiredPriority@type* Property 135
Table B.8 — Output Control Properties 135
Table B.9 — *desiredRecordQuality* Example 138
Table B.10 — Allowed values for the *desiredRecordQuality* Property 139
Table B.11 — Allowed values for the *desiredRecordQuality@type* Property 140
Table B.12 — Content Identification Related Properties 140
Table B.13 — Allowed values for the *scheduledChannelID@type* Property 142
Table B.14 — Matching Content Criteria Properties 145
Table B.15 — Allowed values for the *matchingName@type* Property 146
Table B.16 — Allowed values for the *matchingID@type* Property 147
Table B.17 — Matching Qualifying Criteria Properties 147
Table B.18 — Allowed values for the *matchingRatingLimit* Property Using the MPAA Rating System (*matchingRatingLimit@type* = "*MPAA.ORG*") 150
Table B.19 — Allowed values for the *matchingRatingLimit* Property Using the RIAA Rating System (*matchingRatingLimit@type* = "*RIAA.ORG*") 150
Table B.20 — Allowed values for the *matchingRatingLimit* Property Using the ESRB Rating System (*matchingRatingLimit@type* = "*ESRB.ORG*") 151
Table B.21 — Allowed values for the *matchingRatingLimit* Property Using the TVGUIDELINES Rating System (*matchingRatingLimit@type* = "*TVGUIDELINES.ORG*") 151
Table B.22 — Allowed values for the *matchingRatingLimit@type* Property 152
Table B.23 — Allowed values for the *matchingEpisodeType* Property 153
Table B.24 — Content Control Properties 153

© ISO/IEC 2017 – All rights reserved

Table B.25 — Allowed values for the *durationLimit@effect* Property....156
Table B.26 — Storage Related Properties....157
Table B.27 — Schedule State Properties....159
Table B.28 — Allowed values for the *scheduleState* Property....160
Table B.29 — Allowed values for the *scheduleState@currentErrors* Property....160
Table B.30 — Statistics Properties....161
Table B.31 — Task General Properties....162
Table B.32 — Task Content Identification Properties....163
Table B.33 — *recordQuality* Example....167
Table B.34 — Allowed values for the *recordQuality* Property....168
Table B.35 — Task Matched Content Criteria Properties....169
Table B.36 — Task Matched Qualifying Criteria Properties....170
Table B.37 — Task Matched Content Control Properties....171
Table B.38 — State Related Properties....173
Table B.39 — Allowed values for the attributes of the *taskState* Property....174
Table B.40 — Allowed values for the *taskState* Property....175
Table B.41 — Allowed values for the *taskState@phase* Property....176
Table B.42 — Allowed values for the *taskState@xxx* Properties....179
Table C.1 — Class Properties Overview for *recordScheduleParts* usage....189
Table C.2 — Class Properties Overview for *recordSchedule* usage....193
Table C.3 — Class Properties Overview for *recordTask* usage....197
Table C.4 — *object* Base Class Properties....200
Table C.5 — *object.recordSchedule* Base Class Properties....201
Table C.6 — *object.recordSchedule.direct* Class Properties....202
Table C.7 — *object.recordSchedule.direct.manual* Class Properties....203
Table C.8 — *object.recordSchedule.direct.cdsEPG* Class Properties....204
Table C.9 — *object.recordSchedule.direct.cdsNonEPG* Class Properties....206
Table C.10 — *object.recordSchedule.direct.programCode* Class Properties....207
Table C.11 — *object.recordSchedule.query* Class Properties....208
Table C.12 — *object.recordSchedule.query.contentName* Class Properties....209
Table C.13 — *object.recordSchedule.query.contentID* Class Properties....210
Table C.14 — *object.recordTask* Base Class Properties....212

List of Figures

Figure 1 — Creating a new *recordSchedule*....24
Figure 2 — Capability check....25
Figure 3 — Browse *recordSchedule*....25
Figure 4 — Delete a *recordSchedule*....25
Figure 5 — A Full-Featured State Diagram....63
Figure 6 — A Minimal-Implementation State Diagram....69
Figure C.1 — Class hierarchy for the ScheduledRecording service....187

 © ISO/IEC 2017 – All rights reserved

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 29341-20-14 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.

© ISO/IEC 2017 – All rights reserved

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

 © ISO/IEC 2017 – All rights reserved

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.

© ISO/IEC 2017 – All rights reserved

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

 © ISO/IEC 2017 – All rights reserved

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

© ISO/IEC 2017 – All rights reserved

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

 © ISO/IEC 2017 – All rights reserved

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

© ISO/IEC 2017 – All rights reserved

1 Scope

This service definition is compliant with the UPnP Device Architecture version 1.0 [14]. It defines a service type referred to herein as ScheduledRecording service.

The ScheduledRecording service is a UPnP service that allows control points to schedule the recording of content. Generally, this content is broadcast content, but this specification does not limit itself to broadcast content. This service type enables the following functions:

- Create a *recordSchedule* so that it is added to the list of *recordSchedule* instances. Each *recordSchedule* describes user-level recording instructions for the ScheduledRecording service.
- Browse a list of *recordSchedule* instances stored by the ScheduledRecording service.
- Delete a *recordSchedule* so that it is removed from the list of *recordSchedule* instances.
- Browse a list of *recordTask* instances, stored by the ScheduledRecording service. The ScheduledRecording service may create zero or more *recordTask* instances for each *recordSchedule*. A *recordTask* represents a discrete recording operation of a *recordSchedule*.
- Enable or disable individual *recordTask* instances.
- Enable or disable a *recordSchedule*.
- Receive notifications indicating change of *recordSchedule* or *recordTask* list.

The ScheduledRecording service does not require a dependency on any UPnP services other than a co-located ContentDirectory service, which provides the following functions:

- A ContentDirectory service provides channel line-up to allow users to find recordable channels. A control point may use this metadata when creating a *recordSchedule* on a ScheduledRecording service.
- A ContentDirectory service may provide Electronic Program Guide (EPG) features to allow users to find recordable content. A control point may use this metadata when creating a *recordSchedule* on a ScheduledRecording service.
- Contents recorded by the ScheduledRecording service may be exposed by a ContentDirectory service.

The architectural relationship among the different concepts, defined by the ScheduledRecording service can be summarized as follows: A ScheduledRecording service owns a flat (that is: non-nested) list of *recordSchedule* instances, meaning that the ScheduledRecording service may create, destroy, or change *recordSchedule* instances. A *recordSchedule* represents user-level instructions to perform recording operations. Generally, a user constructs his instructions to a ScheduledRecording service via a control point that invokes UPnP actions that affect the list of *recordSchedule* instances. In all cases, the ScheduledRecording service shall be able to describe discrete recording operations for a *recordSchedule* through a list of associated *recordTask* instances. A *recordTask* can only exist with a *recordSchedule* (that is: never orphaned). Thus when a *recordTask* is created by the ScheduledRecording service, its lifetime depends on its parent *recordSchedule*. An individual *recordTask* can be selectively enabled or disabled.
This service template does not address:

- Implementations where the ScheduledRecording service and its associated ContentDirectory service are not co-located in the same device.

© ISO/IEC 2017 – All rights reserved

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.

 © ISO/IEC 2017 – All rights reserved

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.

ISO/IEC 2017 – All rights reserved

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] – *Unicode Technical Standard #10, Unicode Collation Algorithm, version 4.1.0, revision 14*, M. Davis, K. Whistler, May 5, 2005.
Available at: http://www.unicode.org/reports/tr10/tr10-14.html.

[37] – *Unicode Standard Annex #15, Unicode Normalization Forms, version 4.1.0, revision 25*, M. Davis, M. Dürst, March 25, 2005.
Available at: http://www.unicode.org/reports/tr15/tr15-25.html.

 © ISO/IEC 2017 – All rights reserved

[38] – *Unicode Technical Standard #35, Locale Data Markup Language, version 1.3R1, revision 5*,.M. Davis, June 2, 2005.
Available at: http://www.unicode.org/reports/tr35/tr35-5.html.

[39] – *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.

3 Terms, definitions, symbols and abbreviations

For the purposes of this document, the terms and definitions given in [14] and the following 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 Terms specific to ScheduledRecording service

3.2.1 CDS

ContentDirectory service

3.2.2 EPG

Electronic Program Guide

3.2.3 SRS

ScheduledRecording service

3.2.4 CDS object

An object in a ContentDirectory service metadata hierarchy; that is: *item* or *container*.

3.2.5 User Channel

A User Channel is a ContentDirectory service object that exposes the (continuous) content stream of a particular broadcast channel. Usually, the actual channel that the User Channel exposes is determined by the user through some device-specific interaction. Examples are: manual programming of a number of channel presets; invoking of the auto-scan functionality of a device; predefined fixed channel assignments by the device manufacturer.

3.2.6 Channel Group

A Channel Group is a ContentDirectory service *container* that holds a number of User Channel items. Typically, a Channel Group contains User Channel items that are bound to a particular hardware resource. Examples include: a single analog cable TV tuner, a HDTV digital tuner, an AM/FM radio tuner, etc.

3.2.7 Channel Line-up

A service provider-generated list of channels with their associated content provider.

3.2.8 object

A *recordSchedule* or a *recordTask* (see definition of *recordSchedule* and *recordTask* below).

3.2.9 class

As defined in the ContentDirectory service specification, a class is used to assign a type to an object. It also identifies the minimum required set of properties that shall be present on that object and the allowed properties that may be present. Classes are organized in a hierarchy with certain classes being derived from others as in a typical object-oriented system. This specification defines two base classes (*recordSchedule* and *recordTask*) from which all other classes are derived.

3.2.10 object Modification

An object is considered modified when one of its properties (or its list of properties) is modified; that is: added, removed or changed in value (see definition of property below).

3.2.11 *recordSchedule*

A ScheduledRecording service construct that represents a complete set of recording instructions to the service, which allows the service to generate *recordTask* objects as necessary to record the desired content. The creator of the *recordSchedule* object assigns it a specific class, based on the type and complexity of the instructions, used to identify the content.

A *recordSchedule* is represented in XML as an `<item>…</item>` element.

3.2.12 Conflicting *recordSchedule*

A conflicting *recordSchedule* exists when one or more of its associated *recordTask* instances is in conflict with another *recordTask* instance.

 © ISO/IEC 2017 – All rights reserved

3.2.13 *recordTask*

A ScheduledRecording service construct that represents a discrete recording operation of the underlying recording system. A *recordTask* is created by its parent *recordSchedule* and can not be directly created by the user. The parent-child relationship of *recordSchedule* and *recordTask* can be 1-to-zero or more.

A *recordTask* is represented in XML as an `<item>…</item>` element.

3.2.14 Conflicting *recordTask*

A conflicting *recordTask* exists when it overlaps in time with one or more other *recordTask* instances and the ScheduledRecording service has insufficient resources to record all of them. Existing pre-roll and post-roll adjustments (as defined by the *scheduledStartDateTimeAdjust* and *scheduledDurationAdjust* properties) shall be taken into account when determining conflicts.

3.2.15 *recordScheduleParts*

A ScheduledRecording service construct that represents user-level recording instructions to the service, which provide a template to generate complete *recordSchedule* objects. The creator of the *recordScheduleParts* object assigns it a specific class, based on the type and complexity of the instructions, used to identify the content.

A *recordScheduleParts* is represented in XML as an `<item>…</item>` element.

3.2.16 Property-set Data Types

Certain ScheduledRecording service actions use *property-set* arguments that contain information about a set of properties, typically expressed in the form of an *srs XML Document* (for example, the *Elements* argument of the *CreateRecordSchedule()* action). The set of properties that can exist in a *property-set* argument is implementation dependent. Indeed, the set of allowed properties that a particular ScheduledRecording service chooses to implement is vendor dependent.

This specification currently defines three different *property-set* data types:

- *A_ARG_TYPE_RecordSchedule*
- *A_ARG_TYPE_RecordTask*
- *A_ARG_TYPE_RecordScheduleParts*

Although these three types are different, they are very similar in nature and are defined using the same SRS schema [26], which defines all the properties that can ever occur in any of the three *property-set* data types. They differ only in the set of properties that can appear in them and in the values that are allowed for these properties.

3.2.17 Property

A property in the ScheduledRecording service represents a characteristic of an object. Properties are distinguished by their names. The ScheduledRecording service defines two kinds of properties – independent and dependent. Each independent property has zero or more dependent properties associated with it. Independent property names contain no "@" symbol; they could contain an XML namespace prefix (see below for an explanation of the relationship between properties and XML). Each dependent property is associated either with exactly one independent property or directly with a ScheduledRecording service class. The name of a dependent property that is associated with an independent property is the concatenation of three parts: its associated independent property name, the "@" symbol, and a name for the relationship between the two properties' values. The name of a dependent property that is associated directly with a class is just the "@" symbol followed by the relationship name. Their data types and meanings are defined in Annex B.

Even though ScheduledRecording service properties are not XML objects, XML is used to express them in all exchanges between a control point and a ScheduledRecording service implementation. This creates an unavoidable relationship between XML syntax and property names and values. In XML, an independent property is represented as an element. The property name is used as the element name. The property value is the element content. A dependent property is represented as an attribute in XML. The dependent property's relationship name is used as the attribute name. The dependent property's value is the attribute value. For dependent properties that are associated with an independent property, the attribute appears in the start tag of the element that represents its associated independent property. For dependent properties that are associated directly with a class, the attribute appears in the top-level start tag for each object of that class.

Examples:

Table 1 — Properties in XML

Property Name	XML Representation (srs declared as default namespace)
title	`<title>…</title>`
taskProgramCode	`<taskProgramCode>…</taskProgramCode>`
taskProgramCode@type	`<taskProgramCode type="…">…</taskProgramCode>`
@id	`<item id="…">…</item>`

3.2.18 Member Property

A property is a member of a particular class when the property is declared to be either required or allowed for that class.

3.2.19 Supported Member Property

A supported member property is a member property that is supported by a particular ScheduledRecording service implementation, according to the information returned by the *GetPropertyList()* action.

3.2.20 Multi-valued property

Some independent properties are multi-valued. This means that the property may occur more than once in an object.

3.2.21 Single-valued property

Most independent properties are single-valued. This means thatthe property shall occur at most once in an object. Some single-valued properties can contain a CSV list of values. A dependent property is always considered single-valued, because it can occur at most once with each occurrence of its associated independent property, even though the independent property could be multi-valued.

3.2.22 XML Document

A string that represents a valid XML 1.0 document according to a specific schema. Every occurrence of the phrase "*XML Document*" is preceded by the appropriate root element name, italicized, as listed in column 3, "Valid Root Element(s)" of Table 5.

For example, the phrase "*srs XML Document*" refers to an XML document based on the SRS Schema as defined in [26]. Such a document comprises a single `<srs …>` root element, and it is allowed to be preceded by the XML declaration: `<?xml version="1.0" …?>`.

Therefore, the string containing the *srs XML Document* will have one of the following two forms:

"`<srs …>…</srs>`"
or

 © ISO/IEC 2017 – All rights reserved

```
“<?xml …?>
<srs …>…</srs>”
```

3.2.23 XML Fragment

An *XML Fragment* is a sequence of XML elements that are valid direct or indirect child elements of the root element according to a specific schema. Every occurrence of the phrase “*XML Fragment*” is preceded by the appropriate root element name, italicized, as listed in column 3, “Valid Root Element(s)” of Table 5.

For example, the phrase “*srs XML Fragment*” refers to a sequence of XML elements that are defined in the SRS Schema as defined in [26]:

```
“<item id="…" …>…</item>”
or
“<recordDestination          mediaType="…"          preference="…">
   …
</recordDestination>”
or
“<title>…</title>
 <class>…</class>
 <…>…</…>
 …
 <…>…</…>”
```

3.2.24 actualScheduledStartDateTime

The actual scheduled start date&time of a program item is defined as:

actualScheduledEndDateTime = *scheduledStartDateTime* + *scheduledStartDateTimeAdjust*

where *scheduledStartDateTime* is the scheduled broadcast start date&time of the program item and *scheduledStartDateTimeAdjust* is a user-supplied adjustment to that date&time, for example for pre-roll purposes.

3.2.25 actualStartDateTime

The actual start date&time of a program item is defined as:

actualStartDateTime = *actualScheduledStartDateTime* + any device-specific record startup latency.

3.2.26 actualScheduledEndDateTime

The actual scheduled end time of a program item is defined as:

actualScheduledEndDateTime = *scheduledStartDateTime* + *scheduledDuration* + *scheduledDurationAdjust*

where *scheduledStartDateTime* is the scheduled broadcast start date&time of the program item, *scheduledDuration* is the scheduled broadcast duration of the program item and *scheduledDurationAdjust* is a user-supplied adjustment to that duration, for example to select just a part of the program for recording.

3.2.27 actualEndDateTime

The actual end date&time of a program item is defined as:

actualEndDateTime = *actualScheduledEndDateTime* + any device-specific record teardown latency.

3.2.28 actualScheduledDuration

The actual scheduled duration of a program item is defined as:

actualScheduledDuration = *actualScheduledEndDateTime* – *actualScheduledStartDateTime*

= *scheduledDuration* + *scheduledDurationAdjust* – *scheduledStartDateTimeAdjust*

where *scheduledDuration* is the scheduled broadcast duration of the program item, *scheduledDurationAdjust* is a user-supplied adjustment to that duration, and *scheduledStartDateTimeAdjust* is a user-supplied adjustment to the scheduled start date&time.

3.2.29 Lexical Sort Order

Lexical sort order refers to string sorting – also called collation – based on language and regional conventions. It is *not* based on the binary codes of the characters in strings. Furthermore, lexical sorting is not based on character sets; a single character set may have multiple sort orders, again according to language and regional conventions. It is also possible to have lexical sorts that are further refined according to user preference. For a complete discussion of this topic see [36], and the related standards [37] and [38]. The lexical sort algorithms are defined in [36]. It uses a secondary algorithm defined in [37] and supporting data tables defined in [38]. These three references together – [37], [36] and [38] – should be sufficient to implement a robust lexical sort.

Simple example: one of the most familiar examples is case-insensitive sorting on the ASCII subset of Unicode. In a binary ASCII sort, all lower case letters sort after the upper case “Z” because “Z” has a character code of 0x5A, and all lower case character codes are greater than or equal to 0x61.

More complex example: the “ö” character in German sorts between “n” and “p” characters whereas in Swedish, it sorts after “z”.

3.2.30 Lexical Matching

Lexical matching compares two (sub)strings for equality under certain lexical sorting conditions. *It is important to note* that equality in lexical matching is often less restrictive than equality in lexical sorting. In other words, two strings that are equal under a lexical sort will always be a lexical match. However, two strings that are a lexical match might not be equal under a lexical sort for the same language and region. In some cases, an implementation’s lexical sort might consider all alphabetic characters with diacritical marks (accents, umlauts, circumflexes, etc.) to be distinct, yet the same implementation might ignore diacritical marks in lexical matching. For example, the strings “resumé”, “resume” and “résumé” might sort as “resume” < “resumé” < “résumé”, but when a lexical match using the string “resume”, might find all three strings “resumé”, “resume” and “résumé”. For implementation techniques, see clause 8 in [36].

3.2.31 Simple Non-case-sensitive Sort Order

A simple non-case-sensitive sort order applies only to Roman alphabetic characters. All lower case ASCII alphabetic characters shall sort the same as their uppercase equivalent, except when compared directly with their upper case equivalent, in which case the upper case character should sort before its lower case equivalent. This means that of the following three ordering relations, (1) shall be true, at least one of (2) and (3) shall be true, and (2) should be true.

“A” ≤ “a” < “B” ≤ “b” < … < “Y” ≤ “y” < “Z” ≤ “z” (1)

“A” < “a” < “B” < “b” < … < “Y” < “y” < “Z” < “z” (2)

“A” = “a” < “B” = “b” < … < “Y” = “y” < “Z” = “z” (3)

Additionally, the same upper and lower case relationships should hold for non-ASCII Roman alphabetic characters. That is, lower case alphabetic characters with

 © ISO/IEC 2017 – All rights reserved

diacritical marks should sort as their upper case equivalent, except when compared directly with their upper case equivalent, in which case the upper case character should sort before its lower case equivalent. The ordering relation between ASCII and non-ASCII alphabetic characters is left unspecified. Also, the ordering relation between non-ASCII alphabetic characters that are not upper or lower case equivalents of each other is left unspecified. This is summarized in the following relations, where c represents any non-ASCII Roman alphabetic character. (4) should be true for all c. (5) should be true for all c. If (5) is false for any c, it should be false for all c and (6) should be true for all c.

$$\text{upper}(c) \leq \text{lower}(c) \quad (4)$$

$$\text{upper}(c) < \text{lower}(c) \quad (5)$$

$$\text{upper}(c) = \text{lower}(c) \quad (6)$$

3.2.32 Simple Non-case-sensitive Matching

In a simple non-case-sensitive match, relation (1) above in subclause 3.2.31 shall be true, and relation (6) above in subclause 3.2.31 should be true.

3.2.33 Numeric Sort Order

A sort order in which values are compared numerically. If the type of an individual value is numeric, the numeric value is used. If the type of an individual value is string, the string is converted to a number and that numeric value is used.

3.2.34 Boolean Sort Order

Boolean values are sorted with “0” (false) being less than “1” (true).

3.2.35 Sequenced Sort

A sequenced sort is a sort applied to a set of values, each of which is composed of a sequence of subvalues. The sequence is often in a CSV list, but there are other kinds of sequences used for sorting in this specification. The sequenced sort starts by sorting based on the first subvalue in the sequence. If all values differ in the first subvalue, the sort is finished. Otherwise, each subset of equal subvalues is then sorted based on the next subvalue in the sequence. This process repeats iteratively until there are no more subsets of equal subvalues or the sequence is exhausted.

3.2.36 Sequenced Lexical Sort

A sequenced sort in which all subvalues are strings and the subvalues are compared lexically.

3.2.37 Sequenced Numeric Sort

A sequenced sort in which each subvalue is either a number or the number represented by a string.

3.2.38 Lexical Numeric Sort

A lexical numeric sort is one where one or more substrings are known to represent numbers. The strings are then sorted using a sequenced sort, where the sequence is composed of the sequence of non-numeric and numeric substrings from the larger string.

For example, assume a property has the form <letter>-<number>, where <number> ranges from 1 to 10. In a straight ascending lexical sort, the values “A-10”, “A-1”, “A-2” would sort as: “A-1”, “A-10”, “A-2”. “A-10” sorts before “A-2” because they are equal in the first two character positions, but in the third position, “1” < “2”. However, in a lexical numeric sort, each string is considered to be a sequence of a letter and number separated by a hyphen. These values then sort as “A-1”, “A-2”, “A-10” because all three are equal in the first subvalue, “A”, but the second subvalue sorts as 1, 2, 10 in numeric order.

3.2.39 *type* Relationship Sort

This is a sort defined exclusively for independent properties that have a dependent property relationship named "*type*". These properties are sorted as a sequence of two subvalues: the first subvalue is the value of the property's *xxx@type* dependent property; the second subvalue is the value of the independent property *xxx* itself. The *xxx@type* subvalues are sorted as specified for the dependent *xxx@type* property in its own subclause. The independent property subvalues are sorted according to the order specified in its subclause. Sorting of the independent property may vary with the value of the dependent property.

3.3 Symbols

3.3.1

::
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

 © ISO/IEC 2017 – All rights reserved