

UPOS Ver1.16 RCSD Specification

UnifiedPOS Retail Peripheral Architecture

Version 1.16 RCSD

International Standard
For Implementation of Point Of Service Peripherals

OMG Document Number: ~~retail/2019-04-01~~ [dtd/20-11-03](#)
Original submission date: February 18th, 2019

Standard document URL: <https://www.omg.org/spec/UPOS/>

This proposal adds to and extends the UPOS 1.15 standard.

[Goto Teable1-1](#)

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

IPR Mode

This specification is issued under the RAND Mode ~~based~~ on the OMG IPR Policy.
OMG IPR Policy <https://www.omg.org/cgi-bin/doc.cgi?ipr>

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

[Goto Table 1-2](#)

UPOS Ver1.16 RCSD Specification

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 109 Highland Avenue, Needham, MA 02494, U.S.A.

TRADEMARKS

IMM®, MDA®, Model Driven Architecture®, UML®, UML Cube logo®, OMG Logo®, CORBA® and XMI® are registered trademarks of the Object Management Group, Inc., and Object Management Group™, OMG™, Unified Modeling Language™, Model Driven Architecture Logo™, Model Driven Architecture Diagram™, CORBA logos™, XMI Logo™, CWM™, CWM Logo™, IOP™, MOF™, OMG Interface Definition Language (IDL)™, and OMG SysML™ are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

OMG's Issue Reporting Procedure

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page <http://www.omg.org>, under Documents, Report a Bug/Issue (http://www.omg.org/report_issue.)

Document ~~Submitter~~
VINX Corp.

Document Publishing ~~Supporters~~

OPOS-J
Sorimachi Giken Co. Ltd.
Microsoft Japan Ltd.
SEIKO EPSON Corp.
Toshiba TEC Corp.
Star Micronics Corp.
Fujitsu Frontec Corp.
NCR Corporation
Sharp Corporation
Omron Social Solutions Corp.
NEC Platforms Corp.
Transaction Media Networks Inc.

UPOS Ver1.16 RCSD Specification
TABLE OF CONTENTS

TABLE OF CONTENTS.....	5
PREFACE.....	17
UPOS 1.16 RCSD SPECIFICATION OVERVIEW.....	19
UPDATED ITEMS IN RELEASE 1.16	19
UPDATED ITEMS IN CHAPTER 21 LIGHTS	19
UPDATED ITEMS IN CHAPTER 29 POS POWER	19
ADDED CHAPTERS IN RELEASE 1.16	19
TABLE 1. EDITED ITEMS LISTS FOR THE DTC/20-04-02	20
CAMERA BRIGHTNESS PROPERTY.....	25
CAPASSOCIATEDHARDTOTALSDEVICE PROPERTY.....	25
CAPCAMERA AUTOGAIN PROPERTY.....	25
CAPCAMERA AUTOWHITEBALANCE PROPERTY.....	26
CAPCAMERA BRIGHTNESS PROPERTY.....	26
CAPCAMERA SATURATION PROPERTY.....	28
CAPSTORAGE PROPERTY.....	29
CAPCAMERA VERTICALFLIP PROPERTY.....	29
CAMERA CONTRAST PROPERTY.....	30
CAMERA HORIZONTALFLIP PROPERTY.....	30
PHOTOGRAPH TYPE PROPERTY.....	32
REMAININGRECORDINGTIMEINSEC PROPERTY.....	33
CAMERA VERTICALFLIP PROPERTY.....	33
VIDEOCAPTUREMODE PROPERTY.....	33
VIDEO RECORDINGFRAMERATE PROPERTY.....	35
NOTE: VIDEO CAPTURE DEVICE PROPERTY VALUE RELATIONSHIP.....	37
STOPVIDEO RECORDING METHOD.....	38
TAKEPHOTOGRAPH METHOD.....	38
INDIVIDUALIDS PROPERTY.....	41
INDIVIDUALRECOGNITIONFILTER PROPERTY.....	41
CAPASSOCIATEDHARDTOTALSDEVICE PROPERTY.....	42
CAPSTORAGE PROPERTY.....	42
CHANNEL PROPERTY.....	42
SAMPLINGRATE PROPERTY.....	42
SOUNDTYPE PROPERTY.....	43
STORAGE PROPERTY.....	43
STARTRECORDING METHOD.....	43
STOPRECORDING METHOD.....	44
CAPASSOCIATEDHARDTOTALSDEVICE PROPERTY.....	50
CAPSTORAGE PROPERTY.....	50
CAPVOLUME PROPERTY.....	50
STORAGE PROPERTY.....	50
CAPPOSE PROPERTY.....	55
MOTIONLIST PROPERTY.....	56
CREATEMOTION METHOD.....	56
CAPSTORAGE PROPERTY.....	63
CAPURLFORWARD PROPERTY.....	63
CAPVIDEOTYPE PROPERTY.....	63
DISPLAYMODE PROPERTY.....	63
CAP IMAGETYPELIST PROPERTY.....	64
CANCELURLLOADING METHOD.....	65
GOURLBACK METHOD.....	65
GOURLFORWARD METHOD.....	65
LOADIMAGE METHOD.....	66
LOADURL METHOD.....	66
PLAYVIDEO METHOD.....	66

UPOS Ver1.16 RCSD Specification	
STOPVIDEO METHOD.....	66
UPDATEURLPAGE METHOD.....	66
TABLE 2. CLASS DIAGRAM CHANGE HISTORY TABLE	68
1	68
21/	68
Lights	68
“<<property>> FullColor: boolean” was eliminated since it is no need.....	68
116-1,21	68
C H A P T E R 2 1	73
LIGHTS.....	73
SUMMARY	73
GENERAL INFORMATION	76
CAPABILITIES.....	76
DEVICE SHARING	76
LIGHTS CLASS DIAGRAM.....	77
PROPERTIES(UML ATTRIBUTES)	80
CAPALARM PROPERTY	80
CAPBLINK PROPERTY.....	80
CAPCOLOR PROPERTY	80
CAPFULLCOLOR PROPERTY ADDED IN RELEASE 1.16	80
CAPPATTERN PROPERTY ADDED IN RELEASE 1.16	81
FULLCOLOR PROPERTY ADDED IN RELEASE 1.16	81
MAXLIGHTS PROPERTY	81
METHODS (UML OPERATIONS)	82
SWITCHOFF METHOD	82
SWITCHOFFPATTERN METHOD.....	82
SWITCHON METHOD UPDATED IN RELEASE 1.12-1.16	83
SWITCHONMULTIPLE METHOD ADDED IN RELEASE 1.16	84
SWITCHONPATTERN METHOD ADDED IN RELEASE 1.16	85
EVENTS (UML INTERFACES).....	86
DIRECTIOEVENT.....	86
STATUSUPDATEEVENT	86
C H A P T E R 2 9	87
POS POWER.....	87
SUMMARY	87
GENERAL INFORMATION	91
CAPABILITIES.....	91
DEVICE SHARING	91
MODEL	92
POSPower CLASS DIAGRAM UPDATED IN RELEASE 1.16	93
POSPower STANDBY SEQUENCE DIAGRAM.....	95
POSPower STATE DIAGRAM	96
POSPower POWERSTATE DIAGRAM - PART 1.....	97
POSPower POWERSTATE DIAGRAM - PART 2.....	98
POSPower POWERSTATE DIAGRAM - PART 3.....	99
POSPower STATE CHART DIAGRAM FOR FAN AND TEMPERATURE.....	100
POSPower BATTERY STATE DIAGRAM	101
POSPower POWER TRANSITIONS STATE DIAGRAM.....	102

UPOS Ver1.16 RCSD Specification	
PROPERTIES (UML ATTRIBUTES)	103
BATTERYCAPACITYREMAINING PROPERTY	103
BATTERYCAPACITYREMAININGINSECONDS PROPERTY <i>ADDED IN RELEASE 1.16</i>	103
BATTERYCRITICALLYLOWTHRESHOLD PROPERTY	103
BATTERYCRITICALLYLOWTHRESHOLDINSECONDS PROPERTY <i>ADDED IN RELEASE 1.16</i>	104
BATTERYLOWTHRESHOLD PROPERTY	104
BATTERYLOWTHRESHOLDINSECONDS PROPERTY <i>ADDED IN RELEASE 1.16</i>	104
CAPBATTERYCAPACITYREMAINING PROPERTY	105
CAPBATTERYCAPACITYREMAININGINSECONDS PROPERTY <i>ADDED IN RELEASE 1.16</i>	105
CAPCHARGETIME PROPERTY <i>ADDED IN RELEASE 1.16</i>	105
CAPFANALARM PROPERTY	105
CAPHEATALARM PROPERTY	106
CAPQUICKCHARGE PROPERTY	106
CAPRESTARTPOS PROPERTY	106
CAPSHUTDOWNPOS PROPERTY	106
CAPSTANDBYPOS PROPERTY	107
CAPSUSPENDPOS PROPERTY	107
GOTO TABLE 1-22	107
CAPUPSCHARGESTATE PROPERTY	107
CAPVARIABLEBATTERYCRITICALLYLOWTHRESHOLD PROPERTY	108
CAPVARIABLEBATTERYCRITICALLYLOWTHRESHOLDINSECONDS PROPERTY <i>ADDED IN RELEASE 1.16</i>	108
CAPVARIABLEBATTERYLOWTHRESHOLD PROPERTY	108
CAPVARIABLEBATTERYLOWTHRESHOLDINSECONDS PROPERTY <i>ADDED IN RELEASE 1.16</i>	108
CHARGETIME PROPERTY <i>ADDED IN RELEASE 1.16</i>	109
ENFORCEDSHUTDOWNDELAYTIME PROPERTY	109
POWERFAILDELAYTIME PROPERTY	109
POWERSOURCE PROPERTY	110
QUICKCHARGE MODE PROPERTY	110
QUICKCHARGETIME PROPERTY	110
GOTO TABLE 1-22	111
UPSCHARGESTATE PROPERTY	111
METHODS (UML OPERATIONS)	112
RESTARTPOS METHOD	112
SHUTDOWNPOS METHOD	112
STANDBYPOS METHOD	113
SUSPENDPOS METHOD	113
EVENTS (UML INTERFACES)	114
DIRECTIOEVENT	114
STATUSUPDATEEVENT	114
CHAPTER 39	117
VIDEO CAPTURE	117
SUMMARY	117
GENERAL INFORMATION	123
CAPABILITIES	123
VIDEO CAPTURE CLASS DIAGRAM	124
MODEL	125
MODES	125
Photo shooting mode	125

UPOS Ver1.16 RCSD Specification

Movie shooting mode	126
Input Model	126
Bar Code Scan	126
Individual Recognition	127
DEVICE BEHAVIORS	127
PHOTO MODE.....	127
VIDEO MODE.....	128
DEVICE SHARING	129
PROPERTIES (UML ATTRIBUTES)	130
BARCODEENABLED PROPERTY	130
CAMERAAUTOEXPOSURE PROPERTY	130
CAMERAAUTOFOCUS PROPERTY	130
CAMERAAUTOGAIN PROPERTY	131
CAMERAAUTOWHITEBALANCE PROPERTY	131
CAMERABRIGHTNESS PROPERTY	131
CAPASSOCIATEDHARDTOTALSDEVICE PROPERTY	132
CAPCAMERAAUTOEXPOSITIONURE PROPERTY	132
CAPCAMERAAUTOFOCUS PROPERTY	132
CAPCAMERAAUTOGAIN PROPERTY	132
CAPCAMERAAUTOWHITEBALANCE PROPERTY	132
CAPCAMERABRIGHTNESS PROPERTY	133
CAPCAMERA CONTRAST PROPERTY	133
CAPCAMERAEXPOSURE PROPERTY	133
CAPCAMERAGAIN PROPERTY	133
CAPCAMERAHORIZONTALFLIP PROPERTY	133
CAPCAMERA HUE PROPERTY	134
CAPCAPTURE PROPERTY	134
CAPCAPTURECOLORSPACE PROPERTY	134
CAPCAPTURECOLORSPACELIST PROPERTY	134
CAPCAPTUREFRAMERATE PROPERTY	135
CAPCAPTUREMAXFRAMERATE PROPERTY	135
CAPCAPTURERESOLUTION PROPERTY	135
CAPCAPTURERESOLUTIONLIST PROPERTY	135
CAPDECODED DATA PROPERTY	135
CAPINDIVIDUALRECOGNITION PROPERTY	136
CAPPHOTOGRAPH PROPERTY	136
CAPPHOTOCOLORSPACE PROPERTY	136
CAPPHOTOFRAMERATE PROPERTY	136
CAPPHOTOGRAPHRESOLUTION PROPERTY	137
CAPPHOTOGRAPHRESOLUTIONLIST PROPERTY	137
CAPPHOTOGRAPHTYPE PROPERTY	137
CAPPHOTOGRAPHTYPELIST PROPERTY	137
CAPCAMERASATURATION PROPERTY	138
CAPSTORAGE PROPERTY	138
CAPCAMERAVERTICALFLIP PROPERTY	138
CAPVIDEORECORDING PROPERTY	139
CAPVIDEOCOLORSPACE PROPERTY	139
CAPVIDEORECORDINGFRAMERATE PROPERTY	139
CAPVIDEORECORDINGMAXFRAMERATE PROPERTY	139
CAPVIDEORECORDINGRESOLUTION PROPERTY	140
CAPVIDEORECORDINGRESOLUTIONLIST PROPERTY	140
CAPVIDEORECORDINGTYPE PROPERTY	140
CAPVIDEORECORDINGTYPELIST PROPERTY	140
CAMERA CONTRAST PROPERTY	141
CAMERAEXPOSURE PROPERTY	141
CAMERAGAIN PROPERTY	141
CAMERAHORIZONTALFLIP PROPERTY	142

UPOS Ver1.16 RCSD Specification	
CAMERAHUE PROPERTY	142
CAPTUREPHOTOCOLORSPACE PROPERTY	142
PHOTOCOLORSPACELIST PROPERTY	143
CAPTUREPHOTOFRAMERATE PROPERTY	143
PHOTOMAXFRAMERATE PROPERTY	143
CAPTUREPHOTORESOLUTION PROPERTY	144
PHOTOGRAPHRESOLUTION PROPERTY	144
PHOTORESOLUTIONLIST PROPERTY	144
INDIVIDUALRECOGNITIONENABLED PROPERTY	145
PHOTOGRAPHTYPE PROPERTY	145
PHOTOTYPELIST PROPERTY	145
REMAININGRECORDINGTIMEINSEC PROPERTY	146
CAMERASATURATION PROPERTY	146
STORAGE PROPERTY	147
CAMERAVERTICALFLIP PROPERTY	147
VIDEOCAPTUREMODE PROPERTY	148
VIDEOCOLORSPACE PROPERTY	150
VIDEOCOLORSPACELIST PROPERTY	150
VIDEORECORDINGFRAMERATE PROPERTY	151
VIDEOMAXFRAMERATE PROPERTY	151
VIDEORECORDINGRESOLUTION PROPERTY	151
VIDEORESOLUTIONLIST PROPERTY	152
VIDEORECORDINGTYPE PROPERTY	152
VIDEOTYPELIST PROPERTY	153
NOTE: VIDEO CAPTURE DEVICE PROPERTY VALUE RELATIONSHIP	153
METHODS (UML OPERATIONS)	154
READFRAME METHOD	154
STARTVIDEORECORDING METHOD	155
STOPVIDEORECORDING METHOD	156
TAKEPHOTOGRAPH METHOD	157
EVENTS (UML INTERFACES)	158
DATAEVENT	158
DIRECTIOEVENT	158
ERROREVENT	159
STATUSUPDATEEVENT	161
C H A P T E R 4 0	162
INDIVIDUAL RECOGNITION	162
SUMMARY	162
GENERAL INFORMATION	165
CAPABILITIES	165
INDIVIDUAL RECOGNITION CLASS DIAGRAM	165
MODEL	166
INPUT MODEL	166
DEVICE SHARING	167
INDIVIDUALRECOGNITIONFILTER	168
INDIVIDUALRECOGNITIONFILTER PROPERTY EXAMPLE FORMAT	169
■ Basic Items	169
■ Face Recognition device example	170
INDIVIDUALRECOGNITION INFORMATION PROPERTY EXAMPLE FORMAT	172
■ Basic Items	172
■ Face Recognition Device Example	173

UPOS Ver1.16 RCSD Specification	
PROPERTIES (UML ATTRIBUTES)	174
CAPINDIVIDUALLIST PROPERTY	174
INDIVIDUALIDs PROPERTY	174
INDIVIDUALRECOGNITIONFILTER PROPERTY	174
INDIVIDUALRECOGNITIONINFORMATION PROPERTY	175
EVENTS (UML INTERFACES)	176
DATAEVENT	176
DIRECTIOEVENT.....	176
ERROREVENT	177
STATUSUPDATEEVENT	178
CHAPTER 4 1	179
SOUND RECORDER	179
SUMMARY	179
GENERAL INFORMATION	183
CAPABILITIES.....	183
SOUND RECORDER CLASS DIAGRAM	183
MODEL	184
DEVICE SHARING	186
PROPERTIES(UML ATTRIBUTES)	187
CAPASSOCIATEDHARDTOTALSDEVICE PROPERTY	187
CAPCHANNEL PROPERTY	187
CAPCHANNELLIST PROPERTY	187
CAPSAMPLINGRATE PROPERTY	187
CAPSAMPLINGRATELIST PROPERTY	188
CAPSOUNDTYPE PROPERTY	188
CAPSOUNDTYPELIST PROPERTY	188
CAPSTORAGE PROPERTY	189
CAPRECORDINGLEVEL PROPERTY	189
CHANNEL PROPERTY	189
CHANNELLIST PROPERTY	190
RECORDINGLEVEL PROPERTY	190
REMAININGRECORDINGTIMEINSEC PROPERTY.....	190
SAMPLINGRATE PROPERTY	190
SAMPLINGRATELIST PROPERTY	191
SOUNDATA PROPERTY	191
SOUNDTYPE PROPERTY	191
SOUNDTYPELIST PROPERTY	191
STORAGE PROPERTY	192
METHODS(UML OPERATIONS)	193
STARTRECORDING METHOD	193
STOPRECORDING METHOD	194
EVENTS(UML INTERFACES)	195
DATAEVENT	195
DIRECTIOEVENT.....	195
ERROREVENT	196
STATUSUPDATEEVENT	197
CHAPTER 4 2	198
VOICE RECOGNITION	198

UPOS Ver1.16 RCSD Specification	
SUMMARY	198
GENERAL INFORMATION	202
CAPABILITIES	202
VOICE RECOGNITION CLASS DIAGRAM	202
MODEL	203
DEVICE SHARING	204
PROPERTIES (UML ATTRIBUTES)	205
CAPLANGUAGE PROPERTY	205
HEARINGDATAPATTERN PROPERTY	205
HEARINGDATAWORD PROPERTY	205
HEARINGDATAWORDLIST PROPERTY	206
HEARINGRESULT PROPERTY	207
HEARINGSTATUS PROPERTY	208
LANGUAGELIST PROPERTY	208
METHODS (UML OPERATIONS)	209
STARTHEARINGFREE METHOD	209
STARTHEARINGSENTENCE METHOD	210
STARTHEARINGWORD METHOD	211
STARTHEARINGYESNO METHOD	212
STOPHEARING METHOD	212
EVENTS (UML INTERFACES)	213
DATAEVENT	213
DIRECTIOEVENT	213
ERROR EVENT	214
STATUSUPDATEEVENT	215
CHAPTER 4.3	216
SOUND PLAYER	216
SUMMARY	216
GENERAL INFORMATION	219
CAPABILITIES	219
SOUND PLAYER CLASS DIAGRAM	219
MODEL	220
DEVICE SHARING	221
PROPERTIES(UML ATTRIBUTES)	222
CAPASSOCIATEDHARDTOTALSDEVICE PROPERTY	222
CAPMULTIPLAY PROPERTY	222
CAPSOUNDTYPELIST PROPERTY	222
CAPSTORAGE PROPERTY	223
CAPVOLUME PROPERTY	223
DEVICESOUNDLIST PROPERTY	223
OUTPUTIDLIST PROPERTY	223
STORAGE PROPERTY	224
VOLUME PROPERTY	224
METHODS (UML OPERATIONS)	225
PLAYSOUND METHOD	225
STOPSOUND METHOD	225

UPOS Ver1.16 RCSD Specification	
EVENTS (UML INTERFACES)	226
DIRECTIOEVENT.....	226
ERROREVENT.....	227
OUTPUTCOMPLETEEVENT.....	228
STATUSUPDATEEVENT.....	228
C H A P T E R 4 4	229
SPEECH SYNTHESIS	229
SUMMARY	229
GENERAL INFORMATION	233
CAPABILITIES.....	233
SPEECH SYNTHESIS CLASS DIAGRAM	233
MODEL	234
DEVICE SHARING.....	235
PROPERTIES (UML ATTRIBUTES)	236
CAPLANGUAGE PROPERTY.....	236
CAPPITCH PROPERTY.....	236
CAPSPEED PROPERTY.....	236
CAPVOICE PROPERTY.....	236
CAPVOLUME PROPERTY.....	237
LANGUAGE PROPERTY.....	237
LANGUAGELIST PROPERTY.....	237
OUTPUTIDLIST PROPERTY.....	237
PITCH PROPERTY.....	238
SPEED PROPERTY.....	238
VOICE PROPERTY.....	238
VOICELIST PROPERTY.....	239
VOLUME PROPERTY.....	239
SPEAK METHOD.....	240
SPEAKIMMEDIATE METHOD.....	242
STOPCURRENTSPEAKING METHOD.....	242
STOPSPEAKING METHOD.....	243
EVENTS (UML INTERFACES)	244
DIRECTIOEVENT.....	244
ERROREVENT.....	245
OUTPUTCOMPLETEEVENT.....	246
STATUSUPDATEEVENT.....	246
C H A P T E R 4 5	247
GESTURE CONTROL	247
SUMMARY	247
GENERAL INFORMATION	251
CAPABILITIES.....	251
GESTURE CONTROL CLASS DIAGRAM	251
MODEL.....	252
AUTOMATIC CONTROL.....	252
POSE / MOTION.....	253
DEVICE SHARING.....	253
PROPERTIES (UML ATTRIBUTES)	254
AUTOMODE PROPERTY.....	254

UPOS Ver1.16 RCSD Specification	
AUTOMODELIST PROPERTY	254
CAPASSOCIATEDHARDTOTALSDEVICE PROPERTY	254
CAPMOTION PROPERTY	255
CAPMOTIONCREATION PROPERTY	255
CAPPOSE PROPERTY	255
CAPPOSECREATION PROPERTY	255
CAPSTORAGE PROPERTY	256
JOINTLIST PROPERTY	257
MOTIONLIST PROPERTY	257
POSECREATIONMODE PROPERTY	258
POSELIST PROPERTY	258
STORAGE PROPERTY	259
TABLE OF GESTURE CONTROL DEVICE LISTED ITEMS IN PROPERTY	260
METHODS (UML OPERATIONS)	260
CREATEMOTION METHOD	261
CREATEPOSE METHOD	261
GETPOSITION METHOD	262
SETPOSITION METHOD	263
SETSPEED METHOD	264
STARTMOTION METHOD	265
STARTPOSE METHOD	266
STOPCONTROL METHOD	266
EVENTS (UML INTERFACES)	267
DIRECTIOEVENT	267
ERROREVENT	268
OUTPUTCOMPLETEEVENT	269
STATUSUPDATEEVENT	269
C H A P T E R 4 6	270
DEVICE MONITOR	270
SUMMARY	270
GENERAL INFORMATION	273
CAPABILITIES	273
DEVICE MONITOR CLASS DIAGRAM	273
MODEL	274
DEVICE SHARING	275
PROPERTIES (UML ATTRIBUTES)	275
DEVICEDATA PROPERTY	275
DEVICELIST PROPERTY	276
MONITORINGDEVICELIST PROPERTY	277
METHODS (UML OPERATIONS)	278
ADDMONITORINGDEVICE METHOD	278
CLEARMONITORINGDEVICES METHOD	280
DELETEMONITORINGDEVICE METHOD	280
GETDEVICEVALUE METHOD	280
EVENTS (UML INTERFACES)	281
DATAEVENT	281
DIRECTIOEVENT	281
ERROREVENT	282
STATUSUPDATEEVENT	283

UPOS Ver1.16 RCSD Specification	
C H A P T E R 4 7	284
GRAPHIC DISPLAY	284
SUMMARY	284
GENERAL INFORMATION	288
CAPABILITIES	288
GRAPHICS DISPLAY CLASS DIAGRAM	289
MODEL	290
Image Display Mode	290
Movie Video Display Mode	290
Web Display Mode	292
DEVICE SHARING	292
PROPERTIES (UML ATTRIBUTES)	293
BRIGHTNESS PROPERTY	293
CAPASSOCIATEDHARDTOTALSDEVICE PROPERTY	293
CAPBRIGHTNESS PROPERTY	293
CAPIMAGETYPE PROPERTY	293
CAPSTORAGE PROPERTY	294
CAPURLBACK PROPERTY	294
CAPURLFORWARD PROPERTY	294
CAPVIDEOTYPE PROPERTY	295
CAPVOLUME PROPERTY	295
DISPLAYMODE PROPERTY	296
IMAGETYPE PROPERTY	297
CAP IMAGETYPELIST PROPERTY	297
LOADSTATUS PROPERTY	297
STORAGE PROPERTY	298
URL PROPERTY	298
VIDEOTYPE PROPERTY	299
CAP VIDEOTYPELIST PROPERTY	299
VOLUME PROPERTY	299
METHODS (UML OPERATIONS)	300
CANCELURLLOADING METHOD	300
GOURLBACK METHOD	300
GOURLFORWARD METHOD	300
LOADIMAGE METHOD	301
LOADURL METHOD	301
PLAYVIDEO METHOD	302
STOPVIDEO METHOD	302
UPDATEURLPAGE METHOD	303
EVENTS (UML INTERFACES)	304
DIRECTIOEVENT	304
ERROR EVENT	304
OUTPUTCOMPLETEEVENT	305
STATUSUPDATEEVENT	306
A P P E N D I X K	307
RELATIONSHIP TO OTHER OMG SPECIFICATION AND ACTIVITIES	307
ROBOTICS DOMAIN TASK FORCE	307
Activities in Robotics Domain Task Force	307
RoS Specification	307
Scope of RoS specification	307

UPOS Ver1.16 RCSD Specification	
Robot Service Ontology [RoSO] RFP	308
INTEROPERABILITY BETWEEN UPOS RCSD AND RoIS.....	309
Relationship between UPOS RCSD and RoIS	309
DOCUMENT HISTORY	311
VERSION HISTORY	311
GLOSSARY.....	312
UPOS 1.16 RCSD ISSUES TABLE.....	313

[GotoTable1-4](#)

UPOS Ver1.16 RCSD Specification



OBJECT MANAGEMENT GROUP®

Date: April 2020

Unified POS RCSD, v1.16

FTF Beta 1

This specification adds to and extends the UPOS 1.15 specification.

OMG Document Number: dtc/20-04-02

Normative reference: <https://www.omg.org/spec/UPOS/>

<https://www.omg.org/spec/UPOS/20200301/DeviceMonitorClassDiagram.xmi>
<https://www.omg.org/spec/UPOS/20200301/GestureControlClassDiagram.xmi>
<https://www.omg.org/spec/UPOS/20200301/GraphicDisplayClassDiagram.xmi>
<https://www.omg.org/spec/UPOS/20200301/IndividualRecognitionClassDiagram.xmi>
<https://www.omg.org/spec/UPOS/20200301/LightsClassDiagram.xmi>
<https://www.omg.org/spec/UPOS/20200301/POSPowerClassDiagram.xmi>
<https://www.omg.org/spec/UPOS/20200301/SoundPlayerClassDiagram.xmi>
<https://www.omg.org/spec/UPOS/20200301/SoundRecorderClassDiagram.xmi>
<https://www.omg.org/spec/UPOS/20200301/SpeechSynthesisClassDiagram.xmi>
<https://www.omg.org/spec/UPOS/20200301/VideoCaptureClassDiagram.xmi>
<https://www.omg.org/spec/UPOS/20200301/VoiceRecognitionClassDiagram.xmi>

This OMG document replaces the submission document (retail/2019-06-01, Alpha). It is an OMG Adopted Beta Specification and is currently in the finalization phase. Comments on the content of this document are welcome, and should be directed to issues@omg.org by October 25, 2019.

You may view the pending issues for this specification from the OMG revision issues web page <https://issues.omg.org/issues/lists>.

The FTF Recommendation and Report for this specification will be published in July 2020. If you are reading this after that date, please download the available specification from the OMG Specifications Catalog.

UPOS Ver1.16 RCSD Specification

Preface

OMG

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable, and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies, and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets. More information on the OMG is available at <http://www.omg.org/>.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. All OMG Specifications are available from the OMG website at:

<http://www.omg.org/spec>

Specifications are organized by the following categories:

Business Modeling Specifications

Middleware Specifications

- 1 CORBA/IIOP
- 2 Data Distribution Services
- 3 Specialized CORBA

IDL/Language Mapping Specifications

Modeling and Metadata Specifications

- 4 UML, MOF, CWM, XMI
- 5 UML Profile

Modernization Specifications

Platform Independent Model (PIM), Platform Specific Model (PSM), Interface Specifications

- 6 CORBAServices
- 7 CORBAFacilities

OMG Domain Specifications

CORBA Embedded Intelligence Specifications

CORBA Security Specifications

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications,

UPOS Ver1.16 RCSD Specification

available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. at:

OMG Headquarters
109 Highland Avenue
Needham, MA 02494
USA
Tel: +1-781-444-0404
Fax: +1-781-444-0320
Email: pubs@omg.org

Certain OMG specifications are also available as ISO standards. Please consult <http://www.iso.org>

Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

NOTE: Terms that appear in italics are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

Issues

The reader is encouraged to report any technical or editing issues/problems with this specification to http://www.omg.org/report_issue.htm.

UPOS Ver1.16 RCSD Specification

UPOS 1.16 RCSD Specification Overview ~~Overview~~

Updated Items in Release 1.16

Chapter sections 23 and 38 from UPOS1.15 are included with annotations denoting the changes necessary for supporting the addition of the Retail Communications Service Devices. Chapters 39-47 are new chapters for devices being added to UPOS v1. The following is a list of the properties, methods and chapters.

Updated Items in CHAPTER 21 Lights

Properties

~~CapFullColor Property~~

CapPattern Property

~~FullColor Property~~

Methods

~~switchOn Method~~

switchOnMultiple Method

switchOnPattern Method

switchOffPattern Method

Updated Items in CHAPTER 29 POS Power

Properties

CapBatteryCapacityRemainingInSeconds Property

CapVariableBatteryCriticallyLowThresholdInSeconds Property

CapVariableBatteryLowThresholdInSeconds Property

CapChargeTime Property

BatteryCapacityRemainingInSeconds Property

BatteryCriticallyLowThresholdInSeconds Property

BatteryLowThresholdInSeconds Property

ChargeTime Property

~~TimeMode Property~~

Added Chapters in Release 1.16

CHAPTER 39 Video Capture

CHAPTER 40 Individual Recognition

CHAPTER 41 Sound Recorder

CHAPTER 42 Voice Recognition

CHAPTER 43 Sound Player

CHAPTER 44 Speech Synthesis

CHAPTER 45 Gesture Control

CHAPTER 46 Device Monitor

CHAPTER 47 Graphic Display

[Goto Table 1-5](#)

UPOS Ver1.16 RCSD Specification

Table 1. Edited Items Lists for the dtc/20-04-02

Note: If you click the number you can check the actual revised items and can come back here by clicking the Table1-x number in the specification page.) T

No	Chapter/ Device Name	Items to be corrected	Applied Resolution	JIRANo. Issue TableNo.
1	Referencing OMG document number	Referenced doc number correction	1. Corrected from retail/2019-04-01 to dtc/20-04-02	UPOS 116-1,21 Issue1
2	IRP Mode description	Typo correction	1. Corrected “base on” to “based on”.	UPOS 116-1,21 Issue2
3	Document Submitter	Typo correction	1. Corrected from “Sumbitter” to “ Submitter ” and “Supportes” to “ Supporters ”.	UPOS 116-1,21 Issue3
4	Table of Content	Added the the word of “Table of Content”.	Newly added the word of “Table of Content”	UPOS 116-1,21 Issue4
5	UPOS 1.16 RCSD Specification overview.	Typo correction	1. Changed “Overview” to “ Overview ”	UPOS 116-1,21 Issue5
6	21/ Lights	Summary Properties	1. Eliminated CapFullColor, FullColor properties since existing Color property does have the duplicated function and they were no need.	UPOS 116-1,21 116-7 Issue6
7	21/ Lights	Summary Properties	1. Typo was corrected, since AutoDisable, DataCount, DataEventEnabled and OutputID properties were described “Not Supported ”. This was changed to “Not supported.”	UPOS 116-1,21 Issue149
8	21/ Lights	Summary Properties	1. DeviceEnabled property’ capability regarding the “May use after” description was changed from “open claim” to “open & claim” since to make a unified description as UPOS specification.	UPOS 116-1,21 Issue150
9	21/ Lights	Summary Properties	1. CapFullColor and FullColor properties were eliminated, since their supported capabilities were already included in CapColor property.	UPOS 116-1,21 Issue7
10	21/ Lights	Summary Events	1. Description of TransitionEvent was added, since it was missing.	UPOS 116-1,21 Issue151
11	21/ Lights	CapFullColor Property	1. CapFullColor property was eliminated, since its capability was already included in “ CapColor ” property.	UPOS 116-1,21 Issue7
12	21/ Lights	FullColor Property	1. FullColor property was eliminated, since its capability was already included in CapColor property.	UPOS 116-1,21 Issue7
13	21/ Lights	Summary Method switchOn Method	1. Updated version was not 1.16 but 1.12. 2. Since FullColor property was eliminated, then color parameter description was changed accordingly. 3. FullColor property was eliminated and FullColor property was eliminated from “See also” section.	UPOS 116-1,21 Issue6 Issue7
14	21/ Lights	Summary switchOnMultiple Method	1. Since FullColor property was eliminated, then color parameter description was changed accordingly. 2. This method will activate the multiple lights synchronously and this device behavior was added in “Remarks” section. 3. Since FullColor property was eliminated, it was deleted in “See also” section as reference property.	UPOS 116-1,21 Issue8

UPOS Ver1.16 RCSD Specification

15	29/ POS Power	Summary Properties	1. To make the precise Battery type power management CapBatteryCapacityRemainingInSeconds , BatteryCapacityRemainingInSeconds , CapVariableBatteryCriticallyLowThresholdInSeconds , VariableBatteryCriticallyLowThresholdInSeconds , CapVariableBatteryLowThresholdInSeconds and BatteryLowThresholdInSeconds Properties were added. Instead, TimeMode and CapTimeMode property was eliminated since required second level battery management function was added by those newly added properties.	UPOS 116-1,21 Issue10
16	29/ POS Power	Summary Events	1. Description of TransitionEvent was added, as it was not supported, since it was missing.	UPOS 116-1,21 Issue151
17	29/ POS Power	Properties and Events “My use after “item typo correction	The description of “May use after” item of several properties and events were changed from “Not Supported” to “Not supported” since it was a typo. The properties that this changes apply were AutoDisable , DataCount , DataEventEnabled and OutputID . Also, the events that this changes apply were DataEvent , ErrorEvent and OutputCompleteEvent .	UPOS 116-1,21 Issue149
18	29/ POS Power	BatteryCapacityRemaining InSeconds Property	To make more precise battery power handling, BatteryCapacityRemainingInSeconds property was newly added.	UPOS 116-1,21 Issue11
19	29/ POS Power	BatteryCriticallyLow ThresholdInSeconds Property	To make more precise battery power handling, BatteryCriticallyLowThresholdInSeconds property was newly added.	UPOS 116-1,21 Issue11
20	29/ POS Power	BatteryLowThreshold InSeconds Property	To make more precise battery power handling, BatteryLowThresholdInSeconds property was newly added.	UPOS 116-1,21 Issue11
21	29/ POS Power	CapBatteryCapacity RemainingInSeconds Property	To make more precise battery power handling, CapBatteryCapacityRemainingInSeconds property was newly added.	UPOS 116-1,21 Issue11
22 22-1	29/ POS Power	CapTimeMode , TimeMode Properties	Since battery handling time related properties were newly added and current CapTimeMode and TimeMode properties handling function were included in the newly added properties, therefore CapTimeMode and TimeMode properties were eliminated.	UPOS 116-1,21 Issue10
23	29/ POS Power	CapVariableBatteryCriticallyLow ThresholdInSeconds Property	To make more precise battery power handling, CapVariableBatteryCriticallyLowThresholdInSeconds property was newly added.	UPOS 116-1,21 Issue11
24	29/ POS Power	CapVariableBattery LowThresholdInSeconds Property	To make more precise battery power handling, CapVariableBatteryLowThresholdInSeconds property was newly added.	UPOS 116-1,21 Issue11
25	29/ POS Power	DirectIOEvent Event	In DirectIOEvent section, there was a description of Control. For the UPOS spec. it should be device control and it was corrected.	UPOS 116-1,21 Issue34 Issue152
26	39 / Video Capture	Summary	The description of “May use after” items of several properties were changed from “open” to “Not supported” since it was incorrect and DataEvent was not used in this device. The properties that this changes apply are AutoDisable , DataCount and DataEventEnabled . Also, the events that this changes apply are DataEvent , ErrorEvent and OutputCompleteEvent .	UPOS 116-1,21 Issue33
27	39 / Video Capture	Summary	This device will handle the “Hard Totals” device, therefore, CapAssociatedHardTotalsDevice , CapStorage and Storage properties are newly added.	UPOS 116-1,21 Issue144
28	39 /	Summary	To make property and method name shorten,	UPOS

UPOS Ver1.16 RCSD Specification

	Video Capture		<p>CapPhotograpXXX and PhotographXXX are shortened CapPhotoXXX and PhotoXXX. They are, CapPhotograph, CapPhotographType , PhotographType,</p>	<p>116-1,21 Issue15</p>
29	39 / Video Capture	Summary	<p>To make property name shorten, eliminated the camera word from CapCameraXXX properties, since camera was always used in this device and there was no need to use the wording of camera. They are CapCameraAutoExpositionure , CapCameraAutoFocus CapCameraAutoGain, CapCameraAutoWhiteBalance CapCameraBrightness CapCameraContrast CapCameraExposure CapCameraGain CapCameraHorizontalFlip, CapCameraHue CapCameraSaturation CapCameraVerticalFlip</p>	<p>UPOS 116-1,21 Issue44 Issue45 Issue46 Issue47 Issue48 Issue49 Issue50 Issue51 Issue52 Issue53 Issue54 Issue55</p>
30	39 / Video Capture	Summary	<p>To make short property and method name, eliminate the camera word from the CameraXXX properties, since camera was always used in this device and there was no need to use the wording of camera. They are: CameraAutoExpositionure, CameraAutoFocus, CameraAutoGain, CameraAutoWhiteBalance, CameraBrightness, CameraContrast, CameraExposure, CameraGain, CameraHorizontalFlip, CameraHue, CameraSaturation, CameraVerticalFlip</p>	<p>UPOS 116-1,21 Issue41 Issue42 Issue43</p>
31	39 / Video Capture	Summary	<p>CapCaptureXXX properties are redefined to be able to use both Video mode and Photo mode. Therefore, functionalities of CapCaptureXXX properties have been ported to the both VideoXXX and PhotoXXX properties and some were changed, some were are newly added and some were eliminated. In addition, to make the property name shorten, wording of "Recording" was eliminated. Newly added properties are: CapPhotoColorSpace, CapPhotoFrameRate PhotoMaxFrameRate PhotoColorSpaceList PhotoResolutionList, CapVideoColorSpace VideoMaxFrameRate, VideoColorSpace VideoColorSpaceList, VideoResolutionList Changed properties are: CapPhotographResolution, CapturePhotoColorSpace CapturePhotoFrameRate, CapturePhotoResolution CapVideoRecordingFrameRate CapVideoRecordingResolution, VideoRecordingFrameRate, VideoRecordingResolution Eliminated properties are: CapCaptureFrameRate, CapCaptureMaxFramrate, CapCapture, CapCaptureColorSpace, CapCaptureColorSpaceList, CapCaptureResolution, CapCatureResolutonList, CapVideoRecordingMaxFrameRate, CapVideoRecordingResolutionList</p>	<p>UPOS 116-1,21 Issue56 Issue57 Issue58 Issue59 Issue60</p>
32	39 / Video Capture	Summary	<p>To make the property name shorten, eliminate the recording wording from CapVideoRecordingXXX and VideoRecordingXXX properties and changed the property name. They are: CapVideoRecordingType, VideoRecordingType,</p>	<p>UPOS 116-1,21 Issue61</p>

UPOS Ver1.16 RCSD Specification

33	39 / Video Capture	Summary	After making sure the video and photo functions, some properties were newly added. They are: PhotoTypeList , VideoTypeList	UPOS 116-1,21 Issue63 Issue64
34	39 / Video Capture	Summary	To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX , XXX and XXXList type of properties. Therefore, those properties are eliminated. They are: CapPhotographResolutionList CapVideoRecordingResolutionList CapPhotographTypeList CapCaptureResolutionList CapCaptureColorSpaceList	UPOS 116-1,21 Issue141
35	39 / Video Capture	Summary	Hydra device handling was reconsidered and concluded as follows. The hydra device related properties are eliminated since they are handled by application as hydra connected device properties and there was no need to be described in this device. They are: BarCodeEnabled , IndividualRecognitionEnabled , CapIndividualRecognition , CapDecodeData	UPOS 116-1,21 Issue35 Issue36 Issue37
36	39 / Video Capture	Summary	To handle the video recording precisely, added the new property to handle the remaining recording time. It was : RemainingRecordingTimeInSec .	UPOS 116-1,21 Issue75
37	39 / Video Capture	Summary	In the “version” section summary method table, there was an incorrect description regarding the clearInput method. Since this method was utilized in this device, it was corrected from “Not supported” to “1.16”.	UPOS 116-1,21 Issue19
38	39 / Video Capture	Summary readFrame Method	The readFrame method was eliminated since its function was ported to other methods and properties and it was eliminated.	UPOS 116-1,21 Issue16
39	39 / Video Capture	Summary startVideoRecording Method	To make the method name shorten, the startVideoRecording method name was shortened as startVideo .	UPOS 116-1,21 Issue17
40	39 / Video Capture	Summary stopVideoRecording Method	To make the method name shorten, the stopVideoRecording method name was shortened as stopVideo .	UPOS 116-1,21 Issue18
41	39 / Video Capture	Summary takePhotograph Method	1. To make the method name shorten, takePhotograph method name was changed to takePhoto . 2. Also, parameter was edited since overwrite parameter should not be int32 but boolean and timeout parameter was newly added to avoid taking photo forever and its value was int32 .	UPOS 116-1,21 Issue77
42	39 / Video Capture	Summary DataEvent Event	In the Events table, DataEvent description was changed from “read-only“ to “Not supported”, since DataEvent was not used in this Device and StatusUpdateEvent will be used instead.	UPOS 116-1,21 Issue78
43	39 / Video Capture	Summary OutputCompleteEvent Event	OutputCompleteEvent description had a typo and it was changed from “Not Supported” to “Not supported”.	UPOS 116-1,21 Issue149
44	39 / Video Capture	Summary TransitionEvent Event	TransitionEvent description was added from Ver. 1.16 and this was missing.	UPOS 116-1,21 Issue151
45	39 / Video Capture	General Information Capabilities	1. Capabilities section description was edited. 2. To make the property and method name shorten, in here the word of photo graph changed to photo. 3. Movie was replaced as wording of Video . 4. Data storage will be either “device host” or “storage	UPOS 116-1,21 Issue20 Issue21

UPOS Ver1.16 RCSD Specification

			device” and it was not ”must support the storage device”. Therefore, changed the description regarding the data storage function. 5. “How to detect the individual face or object” related description and “use of hydra connected device” related descriptions were added.	
46	39 / Video Capture	Model Modes	After the device behavior discussion, changed the device modes. Current idea was 3 modes, that was to say, capture only, photo shooting and movie shooting modes. Revised device behavior was 2 modes. They are Photo mode and Video modes. Now it became very clear 2 modes, therefore, device behavior description under 2 modes are completely changed.	UPOS 116-1,21 Issue22 Issue23 Issue24 Issue25 Issue26
47	39 / Video Capture	Model Input Model	After Video Capture Device behavior discussion, it was decided that not to use the DataEvent but use the StatusUpdateEvent. Therefore, Input Model section was eliminated since this section describe the DataEvent handling. Instead of this description Photo Mode and Video Mode sections were added with StatusUpdateEvent description.	UPOS 116-1,21 Issue27 Issue28 Issue29 Issue30 Issue31 Issue32 Issue34 Issue78
48	39 / Video Capture	Model Bar Code Scan	Bar Code Scanner can be used as hydra device with Video Capture device. The hydra device was handled by the hydra device service and there was no need to be described the hydra connected device information in the target device section. Therefore, Bar Code Scan section was eliminated.	UPOS 116-1,21 Issue13
49	39 / Video Capture	Model Individual Recognition	Individual Recognition can be used as hydra device with Video Capture device. The hydra device was handled by the hydra device service and there was no need to be described the hydra connected device information in the target device section. Therefore, Individual Recognition section was eliminated.	UPOS 116-1,21 Issue14
50	39 / Video Capture	Device behaviors Photo Mode Video Mode	Video Capture Device behaviors regarding Photo Mode and Video Mode descriptions were completely revised.	UPOS 116-1,21 Issue73
51	39 / Video Capture	BarCodeEnabled Property	Bar Code Scanner device can be connected with Video Capture device as hydra connected device. Therefore, there was no need to be described the hydra connected device function in the target device and it was eliminated.	UPOS 116-1,21 Issue13
52	39 / Video Capture	Camera AutoExposure Property	1. To make the property name shorten, “Camera” word was eliminated and Camera AutoExposure property name was changed to AutoExposure property. 2. In “Remarks” section, “ If false, auto expose of camera is disabled. ” was eliminated and “Otherwise, it is false.” was added instead. 3. In “See also” section, CapCameraAutoExposition was changed to CapAutoExposure, since “Exposition” and “Exposure” were mixed and it was decided to be unified as the “Exposure”.	UPOS 116-1,21 Issue44
53	39 / Video Capture	Camera AutoFocus Property	1. To make the property name shorten, “Camera” wording was eliminated and CameraAutoFocus was changed to Camera AutoFocus 2. In the “Remarks” section “ If false auto focus of camera is Disabled. ” was eliminated and added “Otherwise, it is false.” for better wording. 3. In the “See also” section, to make the property name shorten Camera wording was eliminated and Cap Camera AutoFocus was changed to CapAutoFocus.	UPOS 116-1,21 Issue45

UPOS Ver1.16 RCSD Specification

54	39 / Video Capture	Camera AutoGain Property	<p>1. To make the property name shorten, "Camera" wording was eliminated and changed CameraAutoGain to AutoGain.</p> <p>2. In "Remarks" section, "If false, auto gain of camera is disabled;" was eliminated and added the "Otherwise it is false" for better wording.</p> <p>In addition, additional explanation regarding the Gain and AutoGain Property was added as follows: "When this property is true, it is possible to read the value of Gain property. However, it is not possible to write and change the value of Gain property. 55 AutoGain property is false, then, it is possible to read, write and change the value of Gain property."</p> <p>3. In "See also" section, from CapCameraAutoGain Property CameraGain Property, "Camera" wording was eliminated to make property name shorten and changed to CapAutoGain and Gain.</p>	UPOS 116-1,21 Issue46
55	39 / Video Capture	Camera AutoWhiteBalance Property	<p>1. To make the property name shorten eliminated the "Camera" wording, and changed the CameraAutoWhiteBalance property name to AutoWhiteBalance.</p> <p>2. In "Remarks" section, "If false, auto white balance of camera is disabled" was eliminated and added "Otherwise, it is false" for better wording.</p> <p>3. In "See also" section, to make shorten the property name CapCameraAutoWhiteBalance was changed to CapAutoWhiteBalance.</p>	UPOS 116-1,21 Issue47
56	39 / Video Capture	Camera Brightness Property	<p>1. To make shorten the property name, "Camera" wording was eliminated and changed CameraBrightness to Brightness.</p> <p>2. In "See also" section, to make shorten the property name "Camera" wording was eliminated and CapCameraBrightness was changed to CapBrightness.</p>	UPOS 116-1,21 Issue48
57	39 / Video Capture	CapAssociatedHardTotalsDevice Property	<p>To utilize the storage device this property was newly added.</p> <p>It was CapAssociatedHardTotalsDevice property section.</p>	UPOS 116-1,21 Issue144
58	39 / Video Capture	Cap Camera AutoExpositionure Property	<p>1. To make shorten the Video Capture Device property name, eliminated the "Camera" wording.</p> <p>2. In the Video Capture device word of "Exposition" and "Exposure" were mixed and it was decided that to make unified and selected the "Exposure". Therefore, the property name was changed CapCameraAutoExposition to CapAutoExposure.</p> <p>3. In "Remarks" section, wording was improved like this. "If true, can change the auto expositionure of camera can be changed". And "If false cannot change the exposition of camera" was eliminated and "Otherwise, it is false" was added for better wording..</p>	UPOS 116-1,21 Issue44
59	39 / Video Capture	Cap Camera AutoFocus Property	<p>1. To make property name shorten, wording of "Camera" was eliminated and property name CapCameraAutoFocus was changed to CapAutoFocus. .</p> <p>2. In "Remarks" section, "If false, automatic gain change of camera is not possible" was eliminated and "Otherwise, it is false" was added for better wording.</p> <p>3. In "See also" section "Camera" wording was eliminated to make property name shorten and CameraAutoFocus was changed to AutoFocus.</p>	UPOS 116-1,21 Issue45
60	39 / Video Capture	Cap Camera AutoGain Property	<p>1. To make property name shorten wording of "Camera" was eliminated and property name CapCameraAutoGain was changed to CapAutoGain.</p>	UPOS 116-1,21 Issue46

UPOS Ver1.16 RCSD Specification

			<p>2. In “Remarks” section, ”If false, automatic gain change of camera is not possible.” was eliminated and “Otherwise, it is false” was added for better wording.</p> <p>3. In “See also” section “Camera” wording was eliminated to make property name shorten and CameraAutoGain was changed to AutoGain..</p>	
61	39 / Video Capture	CapCamera AutoWhiteBalance Property	<p>1. To make property name shorten , wording of “Camera” was eliminated and property name CapCameraAutoWhiteBalance was changed to CapAutoWhiteBalance.</p> <p>2. In “Remarks” section, ”If false, auto white balance of camera is not possible.” was eliminated and “Otherwise, it is false” was added for better wording.</p> <p>3. In “See also” section “Camera” wording was eliminated to make property name shorten and CameraAutoWhiteBalance was changed to AutoWhiteBalance.</p>	UPOS 116-1,21 Issue47
62	39 / Video Capture	CapCamera Brightness Property	<p>1. To make property name shorten, wording of “Camera” was eliminated and property name CapCameraBrightness was changed to CapBrightness.</p> <p>2. In “Remarks” section, ”If false, the brightness of the camera cannot be changed.” was eliminated and “Otherwise, it is false” was added for better wording.</p> <p>3. In “See also” section “Camera” wording was eliminated to make property name shorten and CameraBrightness was changed to Brightness.</p>	UPOS 116-1,21 Issue48
63	39 / Video Capture	CapCamera Contrast Property	<p>1. To make a property name shorten, wording of “Camera” was eliminated and was changed as CapCameraContrast.</p> <p>2. In “Remarks” section, ”If false, cannot change the contrast of camera.” was deleted and “Otherwise, it is false” was added for better wording.</p> <p>3. In “See also” section “Camera” wording was eliminated to make property name shorten as CameraContrast.</p>	UPOS 116-1,21 Issue49
64	39 / Video Capture	CapCamera Exposure Property	<p>1. To make property name shorten, wording of “Camera” was eliminated and CapCameraExposure was changed to CapExposure.</p> <p>2. In “Remarks” section, ”If false, cannot change the exposure of camera.” was eliminated and “Otherwise, it is false” was added for better wording.</p> <p>3. In “See also” section “Camera” wording was eliminated to make property name shorten and CameraExposure was changed to Exposure..</p>	UPOS 116-1,21 Issue50
65	39 / Video Capture	CapCamera Gain Property	<p>1. To make property name shorten, wording of “Camera” was eliminated and CapCameraGain was changed to CapGain.</p> <p>2. In “Remarks” section, ”If false, cannot change the exposure of camera.” was eliminated and “Otherwise, it is false” was added for better wording.</p> <p>3. In “See also” section “Camera” wording was eliminated to make property name shorten and CameraExposure was changed to Exposure..</p>	UPOS 116-1,21 Issue51
66	39 / Video Capture	CapCamera HorizontalFlip Property	<p>1. To make a property name shorten, wording of “Camera” was eliminated and was changed as CapCameraHorizontalFlip.</p> <p>2. In “Remarks” section, ”If false, cannot change the horizontal flip of camera.” was deleted and “Otherwise, it is false” was added for better wording.</p> <p>3. In “See also” section “Camera” wording was eliminated to make property name shorten as Exposure.</p>	UPOS 116-1,21 Issue52

UPOS Ver1.16 RCSD Specification

67	39 / Video Capture	CapCameraHue Property	<p>1. To make property name shorten, wording of “Camera” was eliminated and CapCameraHue was changed to CapHue.</p> <p>2. In “Remarks” section, “If false, hue of the camera cannot be changed” was eliminated and “Otherwise, it is false” was added for better wording.</p> <p>3. In “See also” section “Camera” wording was eliminated to make property name shorten CapCameraHue was changed to CapHue.</p>	UPOS 116-1,21 Issue53
68	39 / Video Capture	CapCapture Property	<p>CapCaptureXXX and CaptureYYY properties were redefined to be able to use both Video mode and Photo mode. Therefore, some of VideoXXX and PhotoXXX properties are changed and newly defined.</p> <p>And CapCapture property was eliminated.</p>	UPOS 116-1,21 Issue56
69	39 / Video Capture	CapCaptureColorSpace Property	<p>CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. Therefore, some of VideoXXX and PhotoXXX properties were changed and newly defined.</p> <p>And CapCaptureColorSpace property was eliminated.</p>	UPOS 116-1,21 Issue57
70	39 / Video Capture	CapCaptureColorSpaceList Property	<p>CapCaptureXXX and CaptureYYY properties were redefined to be able to use both Video mode and Photo mode. Therefore, some of VideoZZZ and PhotoWWW properties were changed and newly defined.</p> <p>And CapCaptureColorSpaceList property was eliminated.</p>	UPOS 116-1,21 Issue141
71	39 / Video Capture	CapCaptureFrameRate Property	<p>CapCaptureXXX and CaptureYYY properties were redefined to be able to use both Video mode and Photo mode. Therefore, some of VideoZZZ and PhotoWWW properties were changed and newly defined.</p> <p>Therefore, CapCaptureColorSpaceList property was eliminated.</p>	UPOS 116-1,21 Issue58
72	39 / Video Capture	CapCaptureMaxFrameRate Property	<p>CapCaptureXXX and CaptureYYY properties were redefined to be able to use both Video mode and Photo mode. Therefore, some of VideoZZZ and PhotoWWW properties were changed and newly defined.</p> <p>Therefore, CapCaptureMaxFrameRate property was eliminated.</p>	UPOS 116-1,21 Issue141
73	39 / Video Capture	CapCaptureResolutionList Property	<p>CapCaptureXXX and CaptureYYY properties were redefined to be able to use both Video mode and Photo mode. Therefore, some of VideoZZZ and PhotoWWW properties were changed and newly defined.</p> <p>Therefore, CapCaptureResolutionList property was eliminated.</p>	UPOS 116-1,21 Issue141
74	39 / Video Capture	CapDecodeData Property	<p>Hydra device was handled by its device service and there was no need to be described by the to be connected device specification. Therefore CapDecodeData property was eliminated since this was related to the Scanner device.</p>	UPOS 116-1,21 Issue37
75	39 / Video Capture	CapIndividualRecognition Property	<p>Hydra device was handled by its device service and there was no need to be described by the to be connected device specification. Therefore CapIndividualRecognition property was eliminated since this was related to the Individual Recognition device.</p>	UPOS 116-1,21 Issue38 Issue39 Issue40
76	39 / Video Capture	CapPhotograph Property	<p>1. Photograph was changed to Photo to make the property name shorten and CapPhotograph was changed to CapPhoto.</p> <p>2. In “Remarks” section description was completely revised. And after, “If it is true”, “photograph function is supported” was eliminated and added the description “it supports the photo function and can take a photo. And to</p>	UPOS 116-1,21 Issue15 Issue62

UPOS Ver1.16 RCSD Specification

			<p>activate the photo mode, the VideoCaptureMode property value needs to set VCAP_VCMODE_PHOTO” was added. In addition, after “If false” “photograph function is not supported” was eliminated and “it’s not supporting the photo function” was newly added. And “If true, it is possible taking a photograph by calling the takePhotograph method. If false, it is not possible taking a lphotograph” was added.</p> <p>3. In “See also” section to make the property name shorten takePhotograph Method was changed to takePhoto method.</p>	
77	39 / Video Capture	CapPhotoColorSpace Property	<p>CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. Therefore, some of VideoZZZ and PhotoWWW properties are changed and newly defined.</p> <p>And CapPhotoColorSpace property was newly added.</p>	UPOS 116-1,21 Issue57
78	39 / Video Capture	CapPhotoFrameRate Property	<p>CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. Therefore, some of VideoZZZ and PhotoWWW properties are changed and newly defined.</p> <p>And CapPhotoFrameRate property was newly added.</p>	UPOS 116-1,21 Issue58
79	39 / Video Capture	CapPhotographResolution Property	<p>1. To make property name shorten Photograph was changed to Photo and CapPhotographResolution property was changed to CapPhotoResolution property..</p> <p>2. In “Remarks” section completely revised description. After, “If it is true”, “it is possible changing the photograph resolution” was eliminated and “taking photo resolution is handled and can be changed.” was newly added. And “If false, it is not possible changing the photograph resolution.” was eliminated and “Otherwise, it is false.” was added for better wording.</p> <p>3. In “See also” section PhotoResolution property was newly added since this was new property to be referenced.</p>	UPOS 116-1,21 Issue59
80	39 / Video Capture	CapPhotographResolutionList Property	<p>To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX , XXX and XXXList type of properties. Therefore, CapPhotographResolutionList property was eliminated.</p>	UPOS 116-1,21 Issue141
81	39 / Video Capture	CapPhotographType Property	<p>1. Photograph was changed to Photo to make the property name shorten and CapPhotographType property was changed as CapPhotoType property..</p> <p>2. In “Remarks” section totally revised description. If it is true, “photograph type can be changed.” was changed to “photo image format type can be changed.” And “If false, photograph type cannot be changed.” was changed to “Otherwise, it is false.” for better wording.</p>	UPOS 116-1,21 Issue64
82	39 / Video Capture	CapPhotographTypeList Property	<p>To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX , XXX and XXXList type of properties. Therefore, CapPhotographTypeList property was eliminated.</p>	UPOS 116-1,21 Issue141
83	39 / Video Capture	CapCameraSaturation Property	<p>1. To make the property name shorten wording of “Camera” was eliminated and CapCameraSaturation was changed to CapCameraSaturation.</p> <p>2. In “Remarks” section, “If false, cannot change the saturation of camera” was eliminated and “Otherwise, it is false” was added for better wording.</p> <p>3. In “See also” section “Camera” wording was eliminated to make property name shorten and CameraSaturation was changed to Saturation.</p>	UPOS 116-1,21 Issue54

UPOS Ver1.16 RCSD Specification

84	39 / Video Capture	CapStorage Property	This device will handle the “Hard Totals” device, therefore, CapStorage properties was newly added.	UPOS 116-1,21 Issue144
85	39 / Video Capture	CapCamera VerticalFlip Property	1. To make the property name shorten wording of “Camera” was eliminated and the CapCamera VerticalFlip property name was changed to CapVerticalFlip . 2. In “Remarks” section, “ If false, cannot change the saturation of camera ” was eliminated and “ Otherwise, it is false ” was added for better wording. 3. In “See also” section “Camera” wording was eliminated to make property name shorten and Camera VerticalFlip was changed to VerticalFlip .	UPOS 116-1,21 Issue55
86	39 / Video Capture	CapVideoRecording Property	1. To make the property name shorten wording of “Recording” was eliminated and CapVideoRecording property was changed to CapVideo . 2. In “Remarks” section, “ If false video recording function is not supported. ” was eliminated and “ Otherwise, it is false ” was added for better wording. And to activate the video mode need to set the VideoCaptureMode property as VCAP_VCMODE_VIDEO and If false this device cannot take a video and recording such kinds of explanation was added. 3. In “See also” section “Recording” wording was eliminated to make property name shorten and StartVideoRecording method name was changed to StartVideo . And VideoCaptureMode property was newly added as reference.	UPOS 116-1,21 Issue56
87	39 / Video Capture	CapVideoColorSpace Property	CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. Therefore, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, CapVideoColorSpace property was newly added.	UPOS 116-1,21 Issue57
88	39 / Video Capture	CapVideoRecording FrameRate Property	1. To make property name shorten, wording of “Recording” was eliminated and CapVideoRecording FrameRate property name was changed to CapVideoFrameRate . 2. In “Remarks” section, after “If true” “ video recording frame rate can be changed ” was eliminated and “ can change the video frame rate from 1 to up to VideoMaxFrameRate property value. ” description was added instead to make clearer VideoFrameRate property value handling. Then, “ If false video recording function is not supported. ” was eliminated and “ Otherwise, it is false ” was added for better wording. 3. In “See also” section for the precise FrameRate handling, VideoMaxFrameRate and VideoFrameRate properties were newly added as reference.	UPOS 116-1,21 Issue58
89	39 / Video Capture	CapVideoRecordingMaxFrameRate Property	CapCaptureXXX properties are redefined to be able to use both Video mode and Photo mode. Therefore, VideoXXX and PhotoXXX properties are some are changed and some are newly added and some are eliminated. Therefore, CapVideoRecordingMaxFrameRate property was eliminated.	UPOS 116-1,21 Issue141
90	39 / Video Capture	CapVideoRecording Resolution Property	1. To make property name shorten wording of “Recording” was eliminated and CapVideoRecording Resolution property name was	UPOS 116-1,21 Issue59

UPOS Ver1.16 RCSD Specification

			<p>changed to CapVideoResolution.</p> <p>2. In “Remarks” section, video recording resolution word was eliminated and replaced by “taking video resolution” and VideoResolutionList property behavior was newly added. In addition, “video recording” was replaced by “taking video”.</p> <p>3. In “See also” section, VideoResolutionList Property, VideoResolution Property were newly added as reference.</p>	
91	39 / Video Capture	CapVideoRecordingResolutionList Property	<p>CapCaptureXXX properties are redefined to be able to use both Video mode and Photo mode. Therefore, VideoXXX and PhotoXXX properties are some are changed and some are newly added and some are eliminated.</p> <p>Therefore, CapVideoRecordingResolutionList property was eliminated.</p>	UPOS 116-1,21 Issue66 Issue141
92	39 / Video Capture	CapVideoRecordingType Property	<p>1. To make property name shorten wording of “Recording” was eliminated and CapVideoRecordingType property name was changed to CapVideoType.</p> <p>2. In “Remarks” section, “video recording” was replaced by “taking video type” and VideoTypeList related description was newly added. In addition, “If false, video recording type cannot be changed” was eliminated and replaced by “Otherwise it is false” for better wording.</p> <p>3. In “See also” section, VideoTypeList property, VideoType property were newly added as reference.</p>	UPOS 116-1,21 Issue67 Issue141
93	39 / Video Capture	CapVideoRecordingTypeList Property	<p>CapCaptureXXX properties are redefined to be able to use both Video mode and Photo mode. Therefore, VideoXXX and PhotoXXX properties are some are changed and some are newly added and some are eliminated.</p> <p>Therefore, CapVideoRecordingTypeList property was eliminated.</p>	UPOS 116-1,21 Issue68 Issue141
94	39 / Video Capture	CameraContrast Property	<p>1. To make property name shorten, wording of “Camera” was eliminated and CameraContrast property name was changed to Contrast.</p> <p>2. In “See also” section “Camera” wording was eliminated to make property name shorten and CapCameraContrast property name was changed to CapContrast.</p>	UPOS 116-1,21 Issue42
95	39 / Video Capture	CameraExposure Property	<p>1. To make property name shorten wording of “Camera” was eliminated and CameraExposure property name was changed to Exposure.</p> <p>2. In “See also” section “Camera” wording was eliminated to make property name shorten and CapCameraExposure property name was changed to CapExposure.</p>	UPOS 116-1,21 Issue42
96	39 / Video Capture	CameraGain Property	<p>1. To make property name shorten wording of “Camera” was eliminated and CameraGain property name was changed to Gain.</p> <p>2. In “Remarks” section, added the new description to explain the AutoGain property behavior to use the value of gain in case when this property is true or false.</p> <p>3. In “See also” section “Camera” wording was eliminated to make property name shorten and CapCameraGain property name was changed to CapGain and AutoGain property was newly added as reference.</p>	UPOS 116-1,21 Issue42
97	39 /	CameraHorizontalFlip Property	<p>1. To make property name shorten, wording of “Camera”</p>	UPOS

UPOS Ver1.16 RCSD Specification

	Video Capture		<p>was eliminated and CameraHorizontalFlip property name was changed to HorizontalFlip.</p> <p>2. In “Remarks” section, added the new description to explain the Horizontal Flip function after “If true, horizontal flip of camera is enabled” added “and it is possible to reverse the camera captured image horizontally” And when it is false “If false, horizontal flip of camera is disabled.” was eliminated and “Otherwise, it is false.” was added for better wording. In addition, descriptions explaining the relationship between this HorizontalFlip and similar function of VerticalFlip were added as “There is a similar property called VerticalFlip property. However, each VerticalFlip property and HorizontalFlip property value can be set independently”</p> <p>3. In “See also” section “Camera” wording was eliminated to make property name shorten and CapCameraHorizontalFlip property name was changed to CapHorizontalFlip. In addition, VerticalFlip and CapVerticalFlip properties were newly added as reference.</p>	116-1,21 Issue42
98	39 / Video Capture	Camera Hue Property	<p>1. To make property name shorten, wording of “Camera” was eliminated and CameraHue property name was changed to Hue.</p> <p>2. In “See also” section “Camera” wording was eliminated to make property name shorten and CapCameraHue property name was changed to CapHue.</p>	UPOS 116-1,21 Issue42
99	39 / Video Capture	Capture PhotoColorSpace Property	<p>1. CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. And, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, CapturePhotoColorSpace property and CapCaptureColorSpaceList property names were changed to PhotoColorSpace and CapPhotoColorSpaceList.</p> <p>2. In “Remarks” section, due to the elimination of readFrame method, device behavior was changed as data is acquired if CapPhotoColorSpace property is true. Also, VideoCaptureMode referencing description was eliminated since mode selection method was changed.</p> <p>3. In “See also” section, readFrame method was eliminated since it was not used in this spec. and CapCapturePhotoColorSpaceList was changed to PhotoColorSpaceList due to the CaptureXXX type property changes and CapPhotoColorSpace property was newly added as reference.</p>	UPOS 116-1,21 Issue56
100	39 / Video Capture	PhotoColorSpaceList Property	<p>1. CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. And, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, PhotoColorSpaceList property was newly added.</p>	UPOS 116-1,21 Issue141
101	39 / Video Capture	Capture PhotoFrameRate Property	<p>1. CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. And, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, CapturePhotoFrameRate property name was changed to PhotoFrameRate property.</p> <p>2. In “Remarks” section, due to the elimination of</p>	UPOS 116-1,21 Issue141

UPOS Ver1.16 RCSD Specification

			<p>readFrame method, device behavior was changed as valid value range from 1 to PhotoMaxFrameRate property. Also, VideoCaptureMode related description was eliminated since it was not used, and “This property is only referenced when VCP_VCM_CAPTURE is set in VideoCaptureMode property.” is eliminated.</p> <p>3. In “See also” section, VideoCaptureMode property and readFrame method names were eliminated since they are not used due to their elimination from the spec. and CapCapturePhotoMaxFrameRate was changed to CapPhotoFrameRate due to the CaptureXXX type property change and PhotoMaxFrameRate property was newly added as reference.</p>	
102	39 / Video Capture	PhotoMaxFrameRate Property	<p>1. CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. And, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, PhotoMaxFrameRate property was newly added.</p>	UPOS 116-1,21 Issue141
103	39 / Video Capture	Cap CapturePhotoResolution Property	<p>1. CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. And, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, CapCapturePhotoResolution property name was changed to PhotoResolution.</p> <p>2. In “Remarks” section, due to the elimination of readFrame method, device behavior was changed and “and the photo taken and recorded with the takePhoto method” was added. Also, VideoCaptureMode related description “This property is only referenced when VCP_VCM_CAPTURE is set in VideoCaptureMode property.” was eliminated since it was not used in this device.</p> <p>3. In “See also” section, VideoCaptureMode property and readFrame method were eliminated since they are not used due to their elimination from the spec. and CapCapturePhotoResolutionList was changed to PhotoResolutionList due to the CaptureXXX type property change and PhotoMaxFrameRate property was newly added as reference.</p>	UPOS 116-1,21 Issue70 Issue141
104	39 / Video Capture	PhotographResolution Property	<p>1. CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. Therefore, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, PhotographResolution property was eliminated.</p>	UPOS 116-1,21 Issue70
105	39 / Video Capture	PhotoResolutionList Property	<p>1. CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. And, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, PhotoResolutionList property was newly added.</p>	UPOS 116-1,21 Issue141
106	39 / Video Capture	IndividualRecognitionEnabled Property	<p>Hydra device handling was reconsidered and concluded as follows. The hydra device related properties are eliminated since they are handled by application as hydra connected device properties and there is no need to be described in this device. Therefore, IndividualRecognitionEnabled property was eliminated.</p>	UPOS 116-1,21 Issue69
107	39 / Video	PhotographType Property	<p>1. To make the property name shorten PhotographType property name was changed to PhotoType.</p>	UPOS 116-1,21 Issue71

UPOS Ver1.16 RCSD Specification

	Capture		<p>2. In the “Remarks” also, CapPhotographTypeList property name was changed to CapPhotoTypeList. Also, VideoCaptureMode property related value description “This property is referenced only when VCP_VCM_PHOTO is set in VideoCaptureMode property.” was eliminated due to the CaptureYYY properties redefinition to be able to use both Video mode and Photo mode.</p> <p>3. In “See also” section, to make the property and method name shorten CapPhotographTypeList Property, takePhotograph Method were changed to CapPhotoTypeList property and takePhoto method.</p>	
108	39 / Video Capture	PhotoTypeList Property	<p>1. CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. And, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, PhotoTypeList property was newly added.</p>	UPOS 116-1,21 Issue64
109	39 / Video Capture	RemainingRecordingTimeInSec Property	<p>To handle the video recording precisely, added the new property to handle the remaining recording time as RemainingRecordingTimeInSec property.</p>	UPOS 116-1,21 Issue75
110	39 / Video Capture	Camera Saturation Property	<p>1. To make shorten the property name wording of “Camera” was eliminated and CameraSaturation property name was changed to Saturation.</p> <p>2. In “See also” section “Camera” wording was eliminated to make property name shorten and CapCameraSaturation property name was changed to CapSaturation.</p>	UPOS 116-1,21 Issue54
111	39 / Video Capture	Storage Property	<p>This device will handle the “Hard Totals” device, therefore, Storage property was newly added.</p>	UPOS 116-1,21 Issue144
112	39 / Video Capture	Camera VerticalFlip Property	<p>1. To make property name shorten wording of “Camera” was eliminated and CameraVerticalFlip property name was changed to VerticalFlip.</p> <p>2. In “Remarks” section, after “If true”, “and it is possible to reverse the video or photo image capturing vertically.” was added and “If false, vertical flipping of camera is disabled.” was replaced by “Otherwise, it is false.” for better wording.</p> <p>In addition, “There is a similar property called HorizontalFlip property and each VerticalFlip property and HorizontalFlip property value can be set independently” was added to explain relationship between VerticalFlip and HorizontalFlip properties.</p> <p>3. In “See also” section “Camera” wording was eliminated to make property name shorten and CapCameraVerticalFlip property name was changed to CapVerticalFlip. In addition, HorizontalFlip property and CapHorizontalFlip property were newly added as referenced properties.</p>	UPOS 116-1,21 Issue43
113	39 / Video Capture	VideoCaptureMode Property	<p>1. Video Capture Device modes are changed from Capture only, Photo shooting and Movie shooting modes to Phot Mode and Video Mode. And, VideoCaptureMode property parameters were changed from 3 modes to 2 modes. Therefore, current capture only mode related VCP_VCMODE_CAPTURE parameter was eliminated.</p> <p>And parameter name VCP was changed to VCAP to fit with other devices and now we have VCAP_VCMODE_PHOTO and VCAP_VCMODE_VIDEO.</p>	UPOS 116-1,21 Issue72

UPOS Ver1.16 RCSD Specification

2. VCAP_VCMODE_VIDEO parameter description was changed taking photograph to taking photo due to the property and method name shorten kinds of things. In addition, added the description related to the data recording and CapPhoto property. That is to say, “and their data recording. Can be set when **CapPhoto** property is true.” And eliminated the color space related description, “~~The values of the **CaptureColorSpace** and **CaptureFrameRate** properties are applied to the color space and frame rate of the frame data that can be acquired by the **readFrame** method, and the resolution is applied to the resolution of the **CapPhotographResolution** property~~”

And then, added the PhotoType related information as follows, “The values of the **PhotoType** property, **PhotoColorSpace** property, **PhotoResolution** property **PhotoFrameRate** property are applied to the taking photo image formats list in the **PhotoTypeList** property, the color space values list in the **PhotoColorSpaceList** property, the resolution values list in the **PhotoResolutionList** property, and the frame rate values within the values of **PhotoMaxFrameRate** property. And taking photo is executed by the **takePhoto** method.”

3. VCAP_VCMODE_VIDEO parameter description was changed like those.

a) “~~This mode is for capture and movie shooting~~” was replaced “This mode is for taking the videos and their data recording. Can be set when **CapVideo** property is true” since mode was changed from Movie shooting to Video and use the CapVideo property.

And color space related description “~~The value of the **CaptureColorSpace** property is applied to the color space of the frame data that can be acquired by the **readFrame** method, the values of the **CapVideoRecordingResolution** property and the **CapVideoRecordingFrameRate** property are applied to the resolution and the frame rate~~” was eliminated and VideoType related description “The value of the **VideoType** property, **VideoColorSpace** property, **VideoResolution** property and **VideoFrameRate** property are applied to the taking video image format list in the **VideoTypeList** property, the color space values list in the **VideoColorSpaceList** property, the resolution values list in the **VideoResolutionList** property and frame rate values within the values of **VideoMaxFrameRate** property. Taking the videos and their data recording will be executed by the **startVideo** method and ends taking the video by using the **stopVideo** method.” was newly added.

4. Initialization description “~~This property is initialized to VCAP_VCMODE_CAPTURE by the open method. Indicate the operation mode of video capture~~” was eliminated and “This property is initialized by the by the **open** method. The default value of this property is VCAP_VCMODE_PHOTO” was newly added.

4. In “See also” section, not used capture and readFrame related properties and methods, ~~**CaptureColorSpace**, **CaptureResolution**, **CaptureFrameRate**, **CapPhotographResolution**, **CapVideoRecordingResolution**, **CapVideoRecordingFrameRate** properties and **readFrame** method~~ were eliminated.

UPOS Ver1.16 RCSD Specification

			And, PhotoColorSpace, VideoColorSpace, PhotoResolution, VideoResolution, VideoFrameRate, PhotoFrameRate, CapPhotoColorSpace, CapVideoColorSpace, CapPhotoResolution, CapVideoResolution, VideoMaxFrameRate, PhotoMaxFrameRate, VideoCaptureMode, CapPhoto, CapVideo, VideoType, VideoTypeList, PhotoType, PhotoTypeList properties and takePhoto, startVideo, stopVideo methods were newly added, as referenced properties and methods.	
114	39 / Video Capture	VideoColorSpace Property	1. CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. And, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, VideoColorSpace property was newly added.	UPOS 116-1,21 Issue141
115	39 / Video Capture	VideoColorSpaceList Property	1. CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. And, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, VideoColorSpaceList property was newly added	UPOS 116-1,21 Issue141
116	39 / Video Capture	VideoRecordingFrameRate Property	1. To make shorten the property name, wording of “Recording” was eliminated and changed VideoRecordingFrameRate property name to VideoFrameRate . 2. In “Remarks” section, readFrame method and movie related description was eliminated and added the video related and CapVideo property related description. That is to say, “frame data acquisition” was replaced “data recording” and also “movie taking” was replaced by “video image capturing and recording”. Utilizing method was changed startVideoRecording to startVideo due to the method name shorten. CapVideo property related description and VideoCaptureMode property related description was added as follows, “and CapVideo property is true. This property is only applied when VCAP_VCMODE_VIDEO is set in VideoCaptureMode property” and, “ This property is only referred when VCP_VCM_VIDEO is set in VideoCaptureMode property ” was eliminated. 3. In “See also” section “Recording” wording was eliminated to make property and method name shorten and CapVideoRecordingMaxFrameRate property, startVideoRecording method were changed to VideoMaxFrameRate and startVideo . readFrame method was eliminated since this was removed from the spec. CapVideo property was newly added as referenced property.	UPOS 116-1,21 Issue73
117	39 / Video Capture	VideoMaxFrameRate Property	1. CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. And, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, VideoMaxFrameRate property was newly added	UPOS 116-1,21 Issue141
118	39 / Video Capture	VideoRecordingResolution Property	1. To make shorten the property name, wording of “Recording” was eliminated and changed VideoRecordingResolution property name to VideoResolution . 2. In “Remarks” section, readFrame method and movie related description was eliminated and added the video	UPOS 116-1,21 Issue141

UPOS Ver1.16 RCSD Specification

			<p>related and CapVideo property related description. That is to say, “the frame data” was replaced “video image data” and “readFrame method” was replaced “Video Capture Device”.</p> <p>startVideoRecording method name was changed to startVideo due to the method name shorten and word of “Recording” was eliminated.</p> <p>Also, CapVideoRecordingResolutionList property name was changed to VideoResolutionList due to the property name shorten and CapXXX and XXX and XXXList relations from the UPOS historical properties hierarchies.</p> <p>And parameter was changed due to the mode changes “This property is only referred when VCP_VCM_VIDEO is set in VideoCaptureMode property.” was eliminated and “This property is only applied when VCAP_VCMODE_VIDEO is set in VideoCaptureMode property and if CapVideo property is true” was newly added as parameter description.</p> <p>3. In “See also” section “Recording” wording was eliminated to make property and method name shorten and UPOS historical CapXXX, XXX and XXXList property relations, CapVideoRecordingMaxFrameRate property, startVideoRecording method were changed to VideoMaxFrameRate and startVideo. readFrame method was eliminated since this was removed from the spec. CapVideo property was newly added as referenced property.</p>	
119	39 / Video Capture	VideoResolutionList Property	<p>1. CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. And, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, VideoResolutionList property was newly added</p>	UPOS 116-1,21 Issue141
120	39 / Video Capture	VideoRecordingType Property	<p>1. To make shorten the property name, wording of “Recording” was eliminated and changed VideoRecordingType property name to VideoType.</p> <p>2. In “Remarks” section, readFrame method and movie related description was eliminated and added the video related and CapVideo property related description. That is to say, “movie taken” was replaced “taking video and recorded”.</p> <p>startVideoRecording method name was changed to startVideo due to the method name shorten and word of “Recording” was eliminated.</p> <p>Also, CapVideoRecordingTypeList property name was changed to VideoTypeList due to the property name shorten and CapXXX and XXX and XXXList relations from the UPOS historical properties hierarchies.</p> <p>And parameter was changed due to the mode changes “This property is only referred when VCP_VCM_VIDEO is set in VideoCaptureMode property.” was eliminated and “This property is only applied when VCAP_VCMODE_VIDEO is set in VideoCaptureMode property and if CapVideo property is true” was newly added as parameter description.</p> <p>3. In “See also” section “Recording” wording was eliminated to make property and method name shorten and UPOS historical CapXXX, XXX and XXXList property relations, CapVideoRecordingTypeList</p>	UPOS 116-1,21 Issue65 Issue141

UPOS Ver1.16 RCSD Specification

			property, startVideoRecording method were changed to VideoTypeList and startVideo . CapVideo , VideoCaptureMode properties were newly added as referenced property.	
121	39 / Video Capture	VideoTypeList Property	1. CapCaptureXXX and CaptureYYY properties are redefined to be able to use both Video mode and Photo mode. And, some of VideoZZZ and PhotoWWW properties are changed and newly defined. Therefore, VideoTypeList property was newly added. 2. Video type related information are listed here to make sure what kinds of video types are used in this device, since this was not very popular for the UPOS users.	UPOS 116-1,21 Issue141 Issue157
122	39 / Video Capture	Note: Video Capture Device Property Value Relationship	To indicate what kinds of properties are related to the video and photo modes, this section was newly added.	UPOS 116-1,21 Not a Issue
123	39 / Video Capture	Methods readFrame -Method	The readFrame method was eliminated since its function was ported to other methods and properties and it was eliminated.	UPOS 116-1,21 Issue74
124	39 / Video Capture	startVideoRecording Method	1. To make the method name shorten wording of “recording” was eliminated and startVideoRecording method name changed to startVideo . 2. In the “Parameter” and “Description” section, movie was replaced by video . 3. In “Remarks” section, due to the so many property name changes, “Recording starts with the setting contents of the CaptureColorSpace and VideoRecordingResolution properties, and recording starts in the format set by the VideoRecordingType property” was eliminated and “Before calling this method, it needs to set the VideoCaptureMode property to VCAP_VCMODE_VIDEO and CapVideo property needs to be true. Video capturing and recording starts with the setting contents of the VideoColorSpace property, VideoResolution property, VideoFrameRate property and VideoType property” was newly added. 125 StatusUpdateEvent handling description was added to make precise device handling as follows, “ StatusUpdateEvent will notify the application that there is a change in the power status or a state change during video capturing and recording.” And due to making the method name shorten stopVideoRecording method name was changed to stopVideo . In addition, “movie execution” was replaced “video capturing and recording” and “movie” was replaced “video” since movie concept was replaced by video in this spec. Also, Storage property was newly added and “the area managed by “Hard Total” service” was replaced “controlled through the Storage Property .” accordingly. 4. In “Errors” section, when it is E_ELLEGAL, additional meaning regarding the VideoCaptureMode property was added as “ VideoCaptureMode property is not VCAP_VCMODE_VIDEO ” 5. In “See also” section, those property names were changed from CaptureVideoColorSpace, VideoRecordingResolution and VideoRecordingType to VideoColorSpace, VideoResolution and VideoType . The stopVideoRecording method name was changed to stopVideo . VideoFrameRate property, StatusUpdateEvent event and VideoCaptureMode property were newly added.	UPOS 116-1,21 Issue75

UPOS Ver1.16 RCSD Specification

<p>125</p>	<p>39 / Video Capture</p>	<p>stopVideoRecording Method</p>	<p>1. To make the method name shorten wording of “Recording” was eliminated and stopVideoRecording method name was changed to stopVideo.</p> <p>2. In “Remarks” section, “video capturing and” was added in front of “recording process” and “recording of the movie image file” was replaced by “taking video”. In addition, synchronously device behavior and StatusUpdateEvent event behavior was newly added as follows, “This method processed synchronously. StatusUpdateEvent will notify the application that there is a change in the power status or a state change during taking video and recording.”</p> <p>3. In “See also” section, startVideoRecording method name change to startVideo, due to the method name shorten and StatusUpdateEvent event name was newly added for precise device status handling.</p>	<p>UPOS 116-1,21 Issue76</p>
<p>126</p>	<p>39 / Video Capture</p>	<p>takePhotograph Method</p>	<p>1. To make the method name shorten “photograph” was replaced “photo” and takePhotograph method name was changed to takePhoto.</p> <p>2. In the overwrite parameter description, there was a mistake, it was not int32 but <i>boolean</i>. And to avoid the taking photo forever timeout parameter was newly added as <i>int32</i>.</p> <p>3. In the “Parameter” section, timeout parameter and related description was newly added as, “Allowed execution time in milliseconds, before the method fails and a timeout ErrorEvent is sent to the application. If FOREVER (-1) the service will wait until a photograph is taken or an application error occurs.”</p> <p>4. In “Remarks” section, due to the property name shorten and CaptureXXX type of property name changes, those property name changes are added. That is to say, CapturePhotoColorSpace, PhotographResolution and PhotographType property names were changed to PhotoColorSpace, PhotoResolution and PhotoType. PhotoFrameRate property was newly added. Add the functional description “take photo”, “and record” Add this method behavior to be used with VideoCaptureMode property, changed the its value VCP_VCM_PHOTO to VCAP_VCMODE_PHOTO and eliminate the “change to the photo shooting mode.” due to the elimination of photo shooting mode. And add the CapPhoto property related description, “this method can be executed if CapPhoto property is true.” And edited the Storage property related description, “image” was replaced by “photo” and “the area managed by “Hard Total” service” was replaced by “controlled through the Storage Property”.</p> <p>5. In “Errors” section, for the E_ELLEGAL value, due to the VideoCaptureMode property behavior change, its value and relation with CapPhoto property was added and edited as follows, “VideoCaptureMode property was not VCAP_VCMODE_PHOTO and CapPhoto property is not true.”</p> <p>6. In “See also” section, due to the property name changes in this section, CapturePhotoColorSpace, PhotographResolution and PhotographType properties names are changed to PhotoColorSpace,</p>	<p>UPOS 116-1,21 Issue77</p>

UPOS Ver1.16 RCSD Specification

			<p>PhotoResolution and PhotoType. In addition, CapPhoto property, PhotoFrameRate Property and StatusUpdateEvent Event were newly added.</p>					
127	39 / Video Capture	DataEvent Event	<p>1. DataEvent is not used in thwas Device and StatusUpdateEvent will be used instead to make precise device control. Therefore DataEvent description was eliminated.</p>	UPOS 116-1,21 Issue78				
128	39 / Video Capture	ErrorEvent Event	<p>1. In “Attribute” section, ErrorResponse description was incorrect since this was not a pointer like pErrorResponse. And “Pointer to the error event response. See ErrorResponse below for values.” was replaced with current UPOS historical description such as “Error Response, whose default value may be overridden by the application. (i.e., this attribute is settable). See <i>ErrorResponse</i> below for values.” And added the E_EXTENDED value related description if there was no room for the storage area. They are as follows, “ If ErrorCode is E_EXTENDED, then ErrorCodeExtended has one of the following values:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>EVCAP_NOROOM</td> <td>The image data storage area does not have enough room to store.”</td> </tr> </tbody> </table> <p>2. In “ErrorLocus attribute” section, EL_INPUT and EL_INPUT_DTA related description was eliminated since DataEvent driven input was not used in this device. 3. In “ErrorResponse attribute” section, in ER_RETRY value’s meaning, “May be valid for some input devices when the locus is EL_INPUT, in which case the input is retried and the error state is exited.” was eliminated since EL_INPUT value was not used in this device. And ER_CLEAR value description, since EL_INPUT was not existing in this device, “EL_INPUT_DATA and EL_OUTPUT” and “This is the default response when the locus is EL_INPUT.” was eliminated. ER_CONTINUEINPUT value related description was eliminated since EL_INPUT was not existing therefore, there was no CONTINUEINPUT kinds of situation. 4. In “Remarks” section, due to the No DataEvent handling situation, DataEvent related description was eliminated. That is to say, “Input error events are not delivered until DataEventEnabled is true, so that proper application sequencing occurs. Unlike a DataEvent, the Device does not disable further DataEvents or input ErrorEvents; it leaves the DataEventEnabled property value at true. Note that the application may set DataEventEnabled to false within its event handler if subsequent input events need to be disabled for a period of time.” was eliminated. 5. In “See also” section, since this was not a device input model device, therefore, ““Device Input Model” on page Intro-22.” was eliminated.</p>	Value	Meaning	EVCAP_NOROOM	The image data storage area does not have enough room to store.”	UPOS 116-1,21 Issue153
Value	Meaning							
EVCAP_NOROOM	The image data storage area does not have enough room to store.”							
129	39 / Video Capture	StatusUpdateEvent Event	<p>1. Instead of DataEvent type input device handling to make precise device handling it is decided to use the StatusUpdateEvent. 2. In the “Description” section added the description of state change as word of “<i>or a state change</i>”</p>	UPOS 116-1,21 Issue79				

UPOS Ver1.16 RCSD Specification

			<p>3. In “Attributes” section, in the status attribute, add the word of “or a state change”</p> <p>As the values of StatusUpdateEvent, those values and related description were newly added, “VCAP_SUE_START_VIDEO_RECORDING, VCAP_SUE_STOP_VIDEO_RECORDING and VCAP_SUE_START_PHOTO”</p> <p>4. In “Remarks” section, new description was added as “Enqueued when the Video Capture Device detects a power state change or a status change.</p> <p>5. In “See Also” section event related description was added, “Events” on page Intro-19.”</p>	
130	40/ Individual Recognition	Summary Method	<p>clearInput and clearInputProperties method were supported in this device, therefore, those version names were changed “Not supported” to “1.16”</p>	UPOS 116-1,21 Issue149
131	40/ Individual Recognition	Summary TransitionEvent Event	<p>TransitionEvent description was added from Ver. 1.16 and this was missing.</p>	UPOS 116-1,21 Issue151
132	40/ Individual Recognition	Model Input Model	<p>1. readValue method description was eliminated since there is no readValue method in this device. That is to say, “The readValue method follows the UnifiedPOS-Input model.” was eliminated. In addition, regarding the DataEvent and IndividualIDs, IndividualRecognitionInformation, CapIndividualList related device behavior description was newly added. They are:</p> <p>“•When an individual is recognized by this device, a DataEvent is delivered to the application after the IndividualIDs property was set to indicate the recognized individuals.” and “•How to recognize the individuals depends on the IndividualRecognitionFilter function, therefore, please refer to the IndividualRecognitionFiler section.</p> <p>•Other device behavior about this device supports the general device input model as listed below.”</p> <p>To make the clear explanation, the location of “•Identifiable individuals are indicated by the CapIndividualList property. Check the functions supported by the device, set validity / invalidity, etc. with the IndividualRecognitionInformation property.” and “Recognized data is stored in the IndividualRecognitionInformation property, IndividualIDs description” were changed. And added the StatusUpdateEvent status handling methodology description was added for the precise device handling, “•The application will be informed about any status change with a StatusUpdateEvent, also all corresponding status properties will be updated before event delivery.”</p>	UPOS 116-1,21 Issue82
133	40/ Individual Recognition	<p>IndividualRecognitionFiler Chapter</p> <ul style="list-style-type: none"> •IndividualRecognitionFilter <p>Example Format</p> <ol style="list-style-type: none"> a) Basic Items b) Face Recognition device example <ul style="list-style-type: none"> •IndividualRecognition Information <p>Property Example Format</p> <ol style="list-style-type: none"> a) Basic Items 	<p>To make the individual Recognition Device handling clearly, those are newly added.</p> <ol style="list-style-type: none"> 1. IndividualRecognitionFilter Chapter <ol style="list-style-type: none"> a) Explain the IndividualRecognitionFilter property behavior 2. IndividualRecognitionFilter Example Format Table <ol style="list-style-type: none"> a) Basic Items Table b) Face Recognition device example Table 3. IndividualRecognition Information Property Example 	UPOS 116-1,21 Issues81

UPOS Ver1.16 RCSD Specification

		b) Face Recognition device example	Format Table a) Basic Items Table b) Face Recognition device example Table	
134	40/ Individual Recognition	CapIndividualList Property	1. In “See also” section, it was not required and “ on pageXX-11 ” was eliminated.	UPOS 116-1,21
135	40/ Individual Recognition	IndividualIDs Property	1. In “Remarks” section, not hold but set was the property function, therefore, “ Holds an IndividualID- recognized by Individual recognition and indicated by separated with a colon (":"). ” was eliminated and “Set the IndividualIDs recognizable Individual recognition device. IndividualIDs values are indicated by separated with a colon (":").” was newly added.	UPOS 116-1,21
136	40/ Individual Recognition	IndividualRecognitionFilter Property	1. In “Remarks” section first paragraph, “ Holds data- indicating the following. ” was eliminated since it was no need. And “Support for various functions” were edited “• Supporting the various functions (Refer to the Individual Recognition Filter Example Format written by JSON and supported function examples)” and “ supported function are defined by the device ” was eliminated since Filter format function Table was newly added and there was no need the eliminated one. In addition, “ Types handled by various functions ” was replaced by “• Various handled function types.” and “Filter setting of various functions.” was replaced by “• Various filter function settings” for better wording. And last paragraph “supporting scope etc.” was changed “supporting scope etc”	UPOS 116-1,21 Issues81
137	40/ Individual Recognition	DataEvent Event	Since DataEvent related description was missing, therefore DataEvent description was newly added.	UPOS 116-1,21 Issue154
138	40/ Individual Recognition	DirectIOEvent Event	Since DirectIOEvent related description was missing, therefore DirectIOEvent description was newly added.	UPOS 116-1,21 Issue154
139	40/ Individual Recognition	ErrorEvent Event	Since ErrorEvent related description was missing, therefore ErrorEvent description was newly added.	UPOS 116-1,21 Issue154
140	40/ Individual Recognition	StatusUpdateEvent Event	Since StatusUpdateEvent related description was missing, therefore StatusUpdateEvent description was newly added.	UPOS 116-1,21 Issue154
141	41/ Sound Recorder	Summary Properties	This device handles the “Hard Totals” device, therefore, CapAssociatedHardTotalsDevice , CapStorage and Storage properties are newly added.	UPOS 116-1,21 Issue144
142	41/ Sound Recorder	Summary Properties	To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX , XXX and XXXList type of properties. Therefore, CapChannellist, CapSamplingRateList and CapSoundTypeList properties were eliminated. And ChannelList , SamplingRateList and SoundTypeList properties were newly added.	UPOS 116-1,21 Issue141
143	41/ Sound Recorder	Summary Properties	To handle the Sound Recording precisely, added the new property to handle the remaining recording time. Therefore, RemainingRecordingTimeInSec property was newly added.	UPOS 116-1,21 Issue86
144	41/ Sound Recorder	Summary Properties	To handle the recording data precisely, newly added the SoundData property.	UPOS 116-1,21
145	41/ Sound Recorder	Summary	clearInput and clearInputProperties method were	UPOS 116-1,21

UPOS Ver1.16 RCSD Specification

	Sound Recorder	Methods	supported in this device, therefore, those version names were changed " Not supported " to "1.16"	Issue149
146	41/ Sound Recorder	Summary Events	TransitionEvent description was added from Ver. 1.16 and this was missing.	UPOS 116-1,21 Issue151
147	41/ Sound Recorder	Summary Events	In this device ErrorEvent 's <i>ErrorResponse</i> was <i>pErrorResponse</i> , this is the pointer of ErrorResponse. Therefore, ErrorResponse in summary was edited as pErrorResponse .	UPOS 116-1,21 Issue154
148	41/ Sound Recorder	General Information Capabilities	To make a device capability precisely, change the description from " Save the recorded sound to a file " to " Record the real-time audio to a file, deliver the recorded sound data to the property that application may read and / or retrieve, and save the recorded sound data file to device memory and / or other storage devices. "	UPOS 116-1,21
149	41/ Sound Recorder	Model	After the huge device model and device behavior discussion. Current device model of Sound Recorder was Device Input Model for event driven input. It is decided not to be an event driven input device but this device will be a device input model in a broad sense. Then this model description was totally revised. Here are the changes that was applied to this model. a) Added the StatusUpdateEvent to handle the real time device control. b) To make the precise device control added the property RemainingRecordingTimeInSec . c) To make the sound data handling added the SoundData property.	UPOS 116-1,21 Issue82
150	41/ Sound Recorder	CapAssociatedHardTotalsDevice Property	This device will handle the "Hard Totals" device, therefore, CapAssociatedHardTotalsDevice property is newly added.	UPOS 116-1,21 Issue144
151	41/ Sound Recorder	CapChannelList Property	To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX , XXX and XXXList type of properties. Therefore, CapChannelList property was eliminated.	UPOS 116-1,21 Issue141
152	41/ Sound Recorder	CapSamplingRateList Property	To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX , XXX and XXXList type of properties. Therefore, CapSamplingRateList property was eliminated.	UPOS 116-1,21 Issue141
153	41/ Sound Recorder	CapSoundTypeList Property	To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX , XXX and XXXList type of properties.	UPOS 116-1,21 Issue141
154	41/ Sound Recorder	CapStorage Property	This device will handle the "Hard Totals" device, therefore, CapStorage property is newly added.	UPOS 116-1,21 Issue144
155	41/ Sound Recorder	Channel Property	Since Channel property value is string, if value description like int32 type is written it is incorrect. Therefore, Value, Meaning and its related descriptions were eliminated in this chapter.	UPOS 116-1,21 Issue83
156	41/ Sound Recorder	Channellist property	To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX , XXX and XXXList type of properties. Therefore, Channellist property was newly added.	UPOS 116-1,21 Issue141
157	41/ Sound Recorder	RemainingRecordingTimeInSec property	To handle the Sound Recording precisely, added the new property to handle the remaining recording time. Therefore, RemainingRecordingTimeInSec property was newly added.	UPOS 116-1,21 Issue86
158	41/ Sound Recorder	SamplingRate Property	Since SamplingRate property value is string, if value description like int32 type is written it is incorrect. Therefore, Value, Meaning and its related descriptions	UPOS 116-1,21 Issue84

UPOS Ver1.16 RCSD Specification

			were eliminated in this chapter.	
159	41/ Sound Recorder	SamplingRateList Property	To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX , XXX and XXXList type of properties. Therefore, SamplingRateList property was newly added.	UPOS 116-1,21 Issue141
160	41/ Sound Recorder	SoundData property	To handle the recording data precisely, newly added the SoundData property.	UPOS 116-1,21 Issue86
161	41/ Sound Recorder	SoundType Property	Since SoundType property value is string, if value description like int32 type is written it is incorrect. Therefore, Value, Meaning and its related descriptions were eliminated in this chapter.	UPOS 116-1,21 Issue85
162	41/ Sound Recorder	SoundTypeList property	To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX , XXX and XXXList type of properties. Therefore, SoundTypeList property was newly added.	UPOS 116-1,21 Issue141
163	41/ Sound Recorder	Storage Property	This device will handle the “Hard Totals” device, therefore, Storage property is newly added.	UPOS 116-1,21 Issue144
164	41/ Sound Recorder	startRecording Method	<p>1. In the parameter of overWrite, there was a description, “return an error” and it is not correct and revised as “raise the UPOSException”</p> <p>2. In the recordingTime parameter, there was a description, “you call” and to make better description it was eliminated and added the “is called” instead.</p> <p>3. In “Remarks” section, current description was not good enough. Therefore, current description, “Recording starts with the settings of the Channel property, SamplingRate property, and RecordingLevel property, and recording starts in the format set by SoundType.” was totally eliminated.</p> <p>And then, added the revised description including the SoundData handling, DataEvent behavior and StatusUpdateEvent handlings. This is newly added and as shown below.</p> <p>“Sound recording starts with the settings of the Channel property, SamplingRate property, and RecordingLevel property and need to set DataEventEnabled property to true. At the same time, recording format setting starts with the SoundType property. When this method is called, if specified recording time is elapsed, recording process will be ended and recorded sound data is provided at the SoundData property that the application may read it and / or process the stored sound data file given as <i>filename</i> argument. When the DataEventEnabled property is true, the DataEvent is enqueued and delivered to the application. StatusUpdateEvent with state SREC_SUE_START_SOUND_RECORDING is evoked when startRecording method is executed to notify the application, the recording has started. When the sound recording is finished, if the specified time of startRecording method has elapsed or stopRecording method has been called, the value of StatusUpdateEvent with state SREC_SUE_STOP_SOUND_RECORDING is evoked to notify the application, the recording has stopped”</p> <p>4. In the “See also” section, referenced items like SoundData property and StatusUpdateEvent event were newly added.</p>	UPOS 116-1,21 Issue86

UPOS Ver1.16 RCSD Specification

165	41/ Sound Recorder	stopRecording Method	<p>1. In “Remarks” section, current description was too simple, “Finish the recording and complete the recording of the audio file.”. This was eliminated and added the new description including the SoundData handling and device controlling by using the StatusUpdateEvent. This is newly added and as shown below.</p> <p>“When this method is called the sound recording process that started by startRecording method is ended and the recording is finished. This method is processed synchronously. After recording and decoding process has been finished, the recorded sound data will be provided at the SoundData property prior to the DataEvent is enqueued, when DataEventEnabled property is true.</p> <p>When stopRecording method is called, a StatusUpdateEvent with status SREC_SUE_STOP_SOUND_RECORDING is evoked to notify the application, the recording has stopped”</p> <p>2. In “See also” section, SoundData property and StatusUpdateEvent event were newly added as reference.</p>	UPOS 116-1,21 Issue87
166	41/ Sound Recorder	DataEvent Event	Since DataEvent related description was missing, therefore DataEvent description was newly added.	UPOS 116-1,21 Issue154
167	41/ Sound Recorder	DirectIOEvent Event	Since DirectIOEvent related description was missing, therefore DirectIOEvent description was newly added.	UPOS 116-1,21 Issue154
168	41/ Sound Recorder	ErrorEvent Event	<p>1.Since ErrorEvent related description was missing, therefore ErrorEvent description was newly added. That is to say, it is “Notifies the application that a Sound Recorder Device error has been detected and suitable response by the application is necessary to process the error condition.”</p> <p>2. In ResultCodeExtended section, EXTENDED value was changed, “ETOF_NOROOM”was changed to “ESRC NOROOM”.</p>	UPOS 116-1,21 Issue88 Issue154
169	41/ Sound Recorder	StatusUpdateEvent Event	Since StatusUpdateEvent related description was missing, therefore StatusUpdateEvent description was newly added.	UPOS 116-1,21 Issue154
170	42/ Voice Recognition	Summary Property	<p>1. HearingDataPattern, HearingDataWord, HearingDataWordList, HearingResult and HearingStatus “May use after” condition was incorrect. They were displayed as “open” but should be “open, claim & enable” They were corrected.</p> <p>2. Common method version regarding clearInput and clear</p>	UPOS 116-1,21 Issue91
171	42/ Voice Recognition	Summary Methods	<p>In the “version” section summary method table, there was an incorrect description regarding the clearInput and clearInputProperties methods.</p> <p>Since this method is utilized in this device, it is corrected from “Not supported” to “1.16” in each common 172hods. .</p>	UPOS 116-1,21 Issue149
172	42/ Voice Recognition	Summary Events	TransitionEvent description was added from Ver. 1.16 and this was missing.	UPOS 116-1,21 Issue151
173	42/ Voice Recognition	Model	<p>1. Description of “control” does not good for the UPOS, since UPOS specification describe the device service. Therefore, control will be changed to “device control”.</p> <p>2. To make a precise device handling, added the StatusUpdateEvent use. Then added the description as follows. “The application will be informed about any status change with a StatusUpdateEvent, also all</p>	UPOS 116-1,21 Issue64 Issue154

UPOS Ver1.16 RCSD Specification

			<p>corresponding status properties will be updated before event delivery.”</p> <p>3 In the Free Recognition chapter, added the description regarding the waiting words as follows. “ It does not specify waiting words and performs voice recognition entrusted to the device.”</p>	
174	42/ Voice Recognition	HearingDataPattern Property	<p>1. This property can be accessed not after open but after open-claim-enable. This was corrected.</p> <p>2. In “Remarks” section, there was a word of “control”. To fit with current UPOS service thinking way changed it to “device control”.</p> <p>3. Also there was the incorrect description DataEvent is “notified” and it was corrected as “enqueued”</p> <p>4. This property can be accessed not after open but after open-claim-enable. This was corrected.</p>	UPOS 116-1,21 Issue91
175	42/ Voice Recognition	HearingDataWord Property	<p>1. This property can be accessed not after open but after open-claim-enable. This was corrected.</p> <p>2. In “Remarks” section, there was the incorrect description DataEvent is “notified” and it was corrected as “enqueued”</p>	UPOS 116-1,21 Issue92
176	42/ Voice Recognition	HearingDataWordList Property	<p>1. This property can be accessed not after open but after open-claim-enable. This was corrected.</p> <p>2. Word list description was not clear and current “”explanation was, “Item:coffee:tea, number:one:two” was eliminated and changed as “item:coffee:tea, count:a:two:three”.</p> <p>3. Also, sentence pattern description was reconsidered for better explanation. Therefore, current description, “Sentence pattern "Pattern01: [product] as [number], Pattern02: as [goods] please" When you recognize the word "one coffee." In the pattern "Pattern 01", "coffee" of the word group "product" and "one" of "number" are recognized. When you recognize the word "one coffee." In the pattern "Pattern01", "coffee" of the word group "product" and "one" of "number" are recognized. At that time, it looks like the following. “Item: coffee, number: one” was totally eliminated and replaced as, “Pattern list: "P1:[count] cup of [item], P2:[item]" startHearingSentence ("en-US", "item:coffee:tea, count:a:two", "P1:[count] cup of [item].P2:[item]") If you speak "Give me two cups of coffee", device recognize “Pattern” as "P1" and “WordList” as “item:coffee, count:two”.</p> <p>The properties are set as follows, HearingDataPattern="P1"; HearingDataWordList="item:coffee, count:two;”</p> <p>4. DataEvent handling was incorrectly described as DataEvent was “notified” and it was corrected as “enqueued”.</p>	UPOS 116-1,21 Issue93
177	42/ Voice Recognition	HearingResult Property	<p>1. This property can be accessed not after open but after open-claim-enable. This was corrected.</p> <p>2. In “Remarks” section, the values of these properties were incorrect. Currently they were TTS_HRESULT_YESNO_YES, TTS_HRESULT_YESNO_NO, TTS_HRESULT_YESNO_CANCEL, TTS_HRESULT_SENTENCE and TTS_HRESULT_FREE. And they were corrected as follows, since Device shorten</p>	UPOS 116-1,21 Issue94

UPOS Ver1.16 RCSD Specification

			<p>name was not TTS but VRCG. TTSVRCG_HRESULT_YESNO_YES, TTSVRCG_HRESULT_YESNO_NO, TTSVRCG_HRESULT_YESNO_CANCEL, TTSVRCG_HRESULT_SENTENCE and TTSVRCG_HRESULT_FREE.</p> <p>3. Regarding the VRCG_HRESULT_YESNO_YES value's meaning description, "finish running voice recognition" method should be replaced by "startHearingYesNo" method, since this value is related to the sartHearingYesNo method. And it is set in the HearingDataWord property not "finish running voice recognition" property but "HearingDataWord" property.</p> <p>4. Regarding the VRCG_HRESULT_YESNO_NO value's meaning description, "finish running voice recognition" method should be replaced by "startHearingYesNo" method, since this value is related to the startHearingYesNo method.</p> <p>5. There was a description DataEvent is "notified". This was incorrect and replaced by "enqueued"</p>	
178	42/ Voice Recognition	HearingStatus Property	<p>1. This property can be accessed not after open but after open-claim-enable. This was corrected.</p> <p>2. In "Remarks" section, the values of these properties were incorrect. Currently they were TTS_HSTATUS_NONE, TTS_HSTATUS_YESNO, TTS_HSTATUS_WORD, TTS_HSTATUS_SENTENCE, TTS_HSTATUS_FREE And they were corrected as follows, since Device shorten name was not TTS but VRCG. VRCG_HSTATUS_NONE, VRCG_HSTATUS_YESNO, VRCG_HSTATUS_WORD, VRCG_HSTATUS_SENTENCE, VRCG_HSTATUS_FREE</p> <p>3. There was a description in the bottom lines of "Remarks" section, property initialization is set by the "control". This should be corrected as "device control".</p>	UPOS 116-1,21 Issue146
179	42/ Voice Recognition	startHearingFree Method	<p>1. In "Remarks" section, current device behavior description, "Device will start waiting without specifying waiting candidates." was replaced by "This method can make a voice recognition from the listed language in the LanguageList property. In addition, this method can be called without specifying the word candidate to be recognized from the application, however recognized word depends on the word recognizing device capability. When this method is called, proper values are set in the HearingDataWord property, HearingResult property and HearingStatus property just before the DataEvent issuing." since current description did not explain the detailed device behavior and the relationship between this method and several properties, DataEvent. In addition, "You can end voice recognition by calling the stopHearing method." was replaced by "Voice recognition ends when stopHearing method is called." for better wording.</p> <p>2. In "See also" section, HearingDataWord, HearingResult, HearingStatus properties were newly</p>	UPOS 116-1,21 Issue95

UPOS Ver1.16 RCSD Specification

<p>180</p>	<p>42/ Voice Recognition</p>	<p>startHearingSentence Method</p>	<p>added as reference.</p> <p>1. Regarding the word information specified in patternList examples were revised to make precise device behavior explanation. Therefore, “For example, in wordList, “Item: coffee:tea, number: one: two” is specified, and a pattern requesting goods and number such as “Two coffee please” and a pattern requesting goods such as “Coffee please” When defining, specify as follows. “Pattern 01: [Number] [Product] Please, Pattern 02: [Product] please” was eliminated and replaced by “Example: You can order coffee or tea. You can also specify how many cups you need. If you want to recognize it by voice, do as follows. Set the startHearingSentence method parameter as follows: WordList: “item:coffee:tea, count:a:two:three” Coffee, Tea -> item:coffee:tea How many cups -> count:a:two:three Invoke the method. startHearingSentence (“en-US”, “item:coffee:tea,count:a:two”, “P1:[count] cup of [item],P2:[item]”) HearingStatus=VRCG_HSTATUS_SENTENCE; People talk to “Give me two cups of coffee” Speech recognition is performed, properties are set, and an event is notified. HearingResult=VRCG_HRESULT_SENTENCE; HearingDataPattern=“P1”; HearingDataWordList=“item:coffee,count:two”; raise DataEvent(0);”, to make a precise device handling.</p> <p>2. In “Remarks” section, current device behavior description, “Start waiting for sentences defined in wordList and patternList.” was replaced by “This method can make a voice recognition from the listed language in the LanguageList property. In addition, this method can recognize the words and sentences that are defined in <i>wordList</i> and <i>patternList</i> as parameter. When this method is called, proper values are set in the HearingDataWord property. HearingResult property and HearingStatus property, just before DataEvent issuing.” since current description did not explain the detailed device behavior and the relationship between this method and several properties, DataEvent. In addition, “You can end voice recognition by calling the stopHearing method.” was replaced by “Voice recognition ends when stopHearing method is called.” for better wording.</p> <p>3. In “See also” section, HearingDataWord, HearingResult, HearingStatus properties were newly added as reference.</p>	<p>UPOS 116-1,21 Issue96</p>
<p>181</p>	<p>42/ Voice Recognition</p>	<p>startHearingWord Method</p>	<p>1. In “Remarks” section, current device behavior description, “Start waiting for sentences defined in wordList.” was replaced by “This method can make a voice recognition from the listed language in the LanguageList property. In addition, this method can recognize the words that are defined in <i>wordList</i> as parameter. When this method is called, proper values are set in the HearingDataWord property. HearingResult property and HearingStatus property, just before DataEvent issuing.” since current description did not</p>	<p>UPOS 116-1,21 Issue97</p>

UPOS Ver1.16 RCSD Specification

			<p>explain the detailed device behavior and the relationship between this method and several properties, DataEvent. In addition, “Application can end voice recognition by calling the stopHearing method.” was replaced by “Voice recognition ends when stopHearing method is called.” for better wording.</p> <p>2. In “See also” section, HearingDataWord, HearingResult, HearingStatus Properties were newly added as reference.</p>	
182	42/ Voice Recognition	startHearingYesNo Method	<p>1. In “Remarks” section, current device behavior description, “Waiting for word candidates corresponding to "Yes" "No" "Cancel" defined by the device is started” was replaced by “This method can make a voice recognition from the listed language in the LanguageList property. In addition, this method can recognize the words that are defined in the device as the recognition candidate corresponding to "Yes" "No" "Cancel". When this method is called, proper values are set in the HearingDataWord property, HearingResult property and HearingStatus property, just before DataEvent issuing.” since current description did not explain the detailed device behavior and the relationship between this method and several properties, DataEvent. In addition, “Application can end voice recognition by calling the stopHearing method.” was replaced by “Voice recognition ends when stopHearing method is called.” for better wording.</p> <p>2. In “See also” section, HearingDataWord, HearingResult, HearingStatus Properties were newly added as reference.</p>	UPOS 116-1,21 Issue98
183	42/ Voice Recognition	stopHearing Method	<p>1. In “Remarks” section, current description “Finish running voice recognition” was eliminated and add, “Voice Recognition ends when this property called.” for better wording. In addition, execution condition description was added as “This method is executed synchronously.”</p>	UPOS 116-1,21 Issue99
184	42/ Voice Recognition	DataEvent Event	<p>Since DataEvent related description was missing, therefore DataEvent description was newly added.</p>	UPOS 116-1,21 Issue154
185	42/ Voice Recognition	DirectIOEvent Event	<p>Since DirectIOEvent related description was missing, therefore DirectIOEvent description was newly added.</p>	UPOS 116-1,21 Issue154
186	42/ Voice Recognition	ErrorEvent Event	<p>1. Since ErrorEvent related description was missing, therefore ErrorEvent description was newly added.</p>	UPOS 116-1,21 Issue154
187	42/ Voice Recognition	StatusUpdateEvent Event	<p>Since StatusUpdateEvent related description was missing, therefore StatusUpdateEvent description was newly added.</p>	UPOS 116-1,21 Issue154
188	43/ Sound Player	Summary Properties	<p>DataCount, DataEventEnabled properties were not supported in this device, therefore, “May use after” times are corrected from “open” to “Not supported”.</p>	UPOS 116-1,21 Issue149
189	43/ Sound Player	Summary Properties	<p>This device handles the “Hard Totals” device, therefore, CapAssociatedHardTotalsDevice, CapStorage and Storage properties are newly added.</p>	UPOS 116-1,21 Issue144
190	43/ Sound Player	Summary Events	<p>TransitionEvent description was added from Ver. 1.16 and this was missing.</p>	UPOS 116-1,21 Issue140 Issue151
191	43/ Sound Player	Model	<p>1. Reconsidering the device handling by using the playSound, stopSound methods, OutputCompleteEvent, StatusUpdateEvent Events,</p>	UPOS 116-1,21 Issue100

UPOS Ver1.16 RCSD Specification

Device handling methodology was completely changed. Therefore, description,

~~“The application calls a startSound method to start playing sound. The Device validates the method parameters and produces an error condition immediately if necessary. If the validation is successful, the Device does the following:~~

- ~~1. Buffers the request in program memory, for delivery to the Physical Device as soon as the Physical Device can receive and process it~~
- ~~2. Sets the **OutputID** property to a unique integer identifier for this request~~
- ~~3. Returns as soon as possible.”~~

~~When the Device successfully completes a request, an **OutputCompleteEvent** is enqueued for delivery to the application. A property of this event contains the **outputID** of the completed request. The application should compare the returned **OutputCompleteEvent** property **OutputID** value with the **OutputID** value set by the asynchronous process method call used to send the data in order to track what data has been successfully sent to the device.”~~ was eliminated.

And, device behavior description, “The Device validates the method parameters and produces an error condition immediately if necessary. If the validation is successful, the Device does the following:” and “Audio files will be played sequentially. When **playSound** method is called, device starts the playing sound that is specified by the method parameters and the requested sound file data placed in a queue and corresponding **OutputID** is stored at **OutputID** property and added to the **OutputIDList** property as a listed value. And sets the **OutputID** property to a unique integer identifier for this request.

- When the sound playing starts **StatusUpdateEvent** is evoked as the value of **SPLY_SUE_START_PLAY_SOUND**.

When the sound playing is finished an **OutputCompleteEvent** is enqueued for the delivery to the application and corresponding **OutputID** is stored in **OutputID** property. At the same time, **StatusUpdateEvent** is evoked as the value of **SPLY_SUE_STOP_PLAY_SOUND**. The application should compare the returned **OutputCompleteEvent** property **OutputID** value with the **OutputID** value set by the asynchronous process method call used to send the data in order to track what data has been successfully sent to the device.

- When **stopSound** method is called, device stop the playing sound according to the **OutputID** property value and the current playing sound is terminated and enqueued sound file data is cleared. After this method is executed, corresponding **OutputID** property and **OutputIDList** values are not changed. No **OutputCompleteEvent** is fired and only **StatusUpdateEvent** will be evoked the value of **SPLY_SUE_STOP_PLAY_SOUND**.” were newly added.

In addition, explaining the use of **StatusUpdateEvent** use and **HardTotals** use descriptions were newly added and current **HardTotals** use explanation was eliminated. That is to say, “The application will be informed about any status change with a **StatusUpdateEvent**, also all

UPOS Ver1.16 RCSD Specification

			<p>corresponding status properties will be updated before event delivery.” was newly added. And “ Applications need to support "hard total" services as audio files played with the startSound method must be placed in the area managed by the "hard total" service.” was eliminated. And “•If device supports either or both of Hard Totals devices and the host file system, the application should set the Storage property accordingly to tell where to access the data file.</p> <p>•If device needs to be able to access the audio files played with playSound method from a Hard Totals device, the CapAssociatedHardTotalsDevice property holds the open name of the associated Hard Totals device.” was newly added.</p>	
192	43/ Sound Player	CapAssociatedHardTotalsDevice Property	This device will handle the “Hard Totals” device, therefore, CapAssociatedHardTotalsDevice property is newly added.	UPOS 116-1,21 Issue144
193	43/ Sound Player	CapStorage Property	This device will handle the “Hard Totals” device, therefore, CapStorage property is newly added.	UPOS 116-1,21 Issue144
194	43/ Sound Player	CapVolume Property	Regarding this property’s access condition, it was described as access after “open- claim ”, however, it is incorrect and this can be accessed after “open”.	UPOS 116-1,21 Issue150
195	43/ Sound Player	OutputIDList Property	Regarding this property’s access condition, it was described as access after “open-claim”, however, it is incorrect and this can be accessed after “open-claim- enable ”.	UPOS 116-1,21 Issue150
196	43/ Sound Player	Storage Property	This device will handle the “Hard Totals” device, therefore, Storage property is newly added.	UPOS 116-1,21 Issue144
197	43/ Sound Player	Volume Property	Regarding this property’s access condition, it was described as access after “open-claim”, however, it is incorrect and this can be accessed after “open-claim- enable ”.	UPOS 116-1,21 Issue101
198	43/ Sound Player	playSound Method	Sound Player device can utilize the sound data either Hard Totals device or device itself. Therefore, description “sound file must be located” was corrected to “sound file might be located”	UPOS 116-1,21 Issue102
199	43/ Sound Player	stopSound Method	In “Remarks” section, playing sound termination timing wording was added as, “ according to the OutputID property value. ”, since this was missing.	UPOS 116-1,21 Issue100
200	43/ Sound Player	DirectIOEvent Event	Since DirectIOEvent related description was missing, therefore DirectIOEvent description was newly added.	UPOS 116-1,21 Issue153
201	43/ Sound Player	ErrorEvent Event	1.Since ErrorEvent related description was missing, therefore ErrorEvent description was newly added.	UPOS 116-1,21 Issue153
202	43/ Sound Player	OutputCompleteEvent Event	Since OutputCompleteEvent related description was missing, therefore OutputCompleteEvent description was newly added.	UPOS 116-1,21 Issue153
203	43/ Sound Player	StatusUpdateEvent Event	Since StatusUpdateEvent related description was missing, therefore StatusUpdateEvent description was newly added.	UPOS 116-1,21 Issue153
204	44/ Speech Synthesis	Summary Properties “My use after “item type correction	The description of “May use after” item of several properties and events were changed from “Not Supported ” to “Not supported” since it is a typo. The properties that this changes apply are DataCount , DataEventEnabled and OutputID .	UPOS 116-1,21 Issue149
205	44/ Speech Synthesis	Summary Event “My use after “item type correction	The description of “May use after” item of several properties and events were changed from “Not Supported ” to “Not supported” since it is a typo.	UPOS 116-1,21 Issue149

UPOS Ver1.16 RCSD Specification

			The Event that this changes apply are DataEvent .	
206	44/ Speech Synthesis	Summary Events	TransitionEvent description was added from Ver. 1.16 and this was missing.	UPOS 116-1,21 Issue151
207	44/ Speech Synthesis	Model	<p>1.General Speech Synthesis device description section did have some to be corrected items. There was a description, “asynchronous output devices:” and this is corrected, “output devices with some enhancements.” since it is better to talk about the Speech Synthesis device.</p> <p>2. Regarding the speak and speakImmediate method explanation, there were typo, “to start speaking the words” and “speakImmediate method ended”. They corrected as “to start speaking from the words” and “speak Immediate method ends” for better wording.</p> <p>3. To make precise and real time device handling, use of StatusUpdateEvent, OutputCompleteEvent, and details of method handling methodology instruction will be required. Then current generic output device handling like “The device validates the method parameters and produces an error condition immediately if necessary. If the validation is successful, the device does the following: <ol style="list-style-type: none"> 1. Buffers the request in program memory, for delivery to the physical device as soon as the physical device can receive and process it. 2. Sets the OutputID property to a unique integer identifier for this request. 3. Returns as soon as possible. When the device successfully completes a request, an OutputCompleteEvent is enqueued for delivery to the application. A property of this event contains the outputID of the completed request. The application should compare the returned OutputCompleteEvent property's OutputID value with the OutputID value set by the asynchronous process method call used to send the data in order to track what data has been successfully sent to the device.” was eliminated.</p> <p>And “When speak or speakImmediate method is called device start the speaking based on the setting value of Language, Volume, Pitch and Speed properties. And requested utterance written by text data placed in a queue and corresponding OutputID is stored at OutputID property and added to the OutputIDList property as listed value. And sets the OutputID property to a unique integer identifier for this request. When an utterance of speak method or speakImmediate method starts, StatusUpdateEvent is evoked as the value of SPSY_SUE_START_SPEAK. When the utterance is finished an OutputCompleteEvent is enqueued for the delivery to the application and corresponding OutputID is stored in OutputID property. At the same time StatusUpDateEvent is evoked as the value of SPSY_SUE_STOP_SPEAK. The application should compare the returned OutputCompleteEvent property OutputID value with OutputID value set by the asynchronous process method call used to send the data in order to track what data has been successfully sent to the device. When speakImmediate method is called during the utterance of speak method or speakImmediate method call, utterance will be stopped</p>	UPOS 116-1,21 Issue104 Issue105 Issue109

UPOS Ver1.16 RCSD Specification

		<p>immediately. And StatusUpdateEvent is evoked as the value of SPSY_SUE_STOP_SPEAK. However, OutputCompleteEvent is not fired. And current speak method or speakImmediate method corresponding OutputID property and OutputIDList property values are not changed. When stopCurrentSpeaking method is called, current utterance generated by speak method or speakImmediate method will be stopped and StatusUpdateEvent is evoked as the value of SPSY_SUE_STOP_SPEAK. And no OutputCompleteEvent is fired. And current speak method or speakImmediate method corresponding OutputID property and OutputIDList property values are not changed. When stopSpeaking method is called, specified OutputID valued utterance is stopped and deleted. And OutputID property value in the OutputIDList property is eliminated. When utterance is stopped StatusUpdateEvent is evoked as the value of SPSY_SUE_STOP_SPEAK. And no OutputCompleteEvent is fired.</p> <p>If an error occurs while processing a request, an ErrorEvent is enqueued which will be delivered to the application after the events already enqueued, including OutputCompleteEvent. No further asynchronous output will occur until the event has been delivered to the application. If the response is ER_CLEAR, then outstanding asynchronous output is cleared. If the response is ER_RETRY, then output is retried; note that if several outputs were simultaneously in progress at the time that the error was detected, then the service may need to retry all of these outputs. Asynchronous output is always performed on a first-in first-out basis. If the request is terminated before completion, due to reasons such as the application calling the clearOutput method, then no OutputCompleteEvent is delivered.” was added.</p> <p>In addition, device status handling will be done mainly by StatusUpdateEvent, therefore, “Application can also delete the output individually by calling the stopCurrentSpeaking, stopSpeaking method. Also in this case OutputCompleteEvent will not be notified.” was eliminated and “The application will be informed about any status change with a StatusUpdateEvent, also all corresponding status properties will be updated before event delivery.” was newly added instead.</p>	
<p>208</p>	<p>44/ Speech Synthesis</p>	<p>speak Method</p>	<p>UPOS 116-1,21 Issue106</p>

1. In “Remarks” section, “~~Device will utter the words specified by Text~~” was eliminated and changed to “Device utters after converting the specified string into speech.”, since uttering will be done based on the string data.

2. Parameter setting is complicated therefore, changed the simple context to the parameter setting Table.

3. Parameter explanation was improved and “~~Tags without reset are specified in the form of “\tag = value \”. For example, when specifying Text as follows, “Hello \pause = 1000 \pitch = 150 \It’s nice weather today \reset \”. “Hello” speaks according to the original setting. Then wait for 1000 milliseconds. “Today” speaks Pitch at 150%. “Nice weather,” I will speak according to the original settings.~~” was eliminated and “If dialogue is “ Hello. Today, it’s nice weather.”

UPOS Ver1.16 RCSD Specification

			<p>Then if you would like to use the default setting of speed, volume, pitch for the “Hello”. And would like to put a pose between “Hello” and “Today” 1000 milliseconds and would like to change the speaking pith of “Today” to 150 and increase the volume to 80. Then for the “It’s nice weather” would like return to the default value by using the reset. It is described as follows Hello. {pause=1000,pitch=150,volume=80}Today, {reset}It’s nice weather.” was newly added for better wording. Also utterance definition Table was newly added. And description supporting the OutputID and OutputCompleteEvent description as “When this method is called by the application, device validate the method parameters, and if validation is successful buffer the request in program memory and deliver it to the device and process it. And device sets the unique integer identifier into the OutputID property. When device successfully complete a request an OutputCompleteEvent is enqueued for delivery to the application.” was newly added. 4. In “See also” section, OutputID Property was newly added as reference.</p>	
209	44/ Speech Synthesis	speakImmediate Method	1. In “See also” section, speak method is newly added as reference.	UPOS 116-1,21 Issue107
210	44/ Speech Synthesis	stopCurrentSpeaking Method	To make the precise device handling, current description “ Stop the currently executed utterance. ” was eliminated and changed to “The speak method and speakImmediate method start the speaking words specified by text and ends when stopCurrentSpeaking method is called.”.	UPOS 116-1,21 Issue108
211	44/ Speech Synthesis	DirectIOEvent Event	Since DirectIOEvent related description was missing, therefore DirectIOEvent description was newly added.	UPOS 116-1,21 Issue154
212	44/ Speech Synthesis	ErrorEvent Event	Since ErrorEvent related description was missing, therefore ErrorEvent description was newly added.	UPOS 116-1,21 Issue154
213	44/ Speech Synthesis	OutputCompleteEvent Event	Since OutputCompleteEvent related description was missing, therefore OutputCompleteEvent description was newly added.	UPOS 116-1,21 Issue154
214	44/ Speech Synthesis	StatusUpdateEvent Event	Since StatusUpdateEvent related description was missing, therefore StatusUpdateEvent description was newly added.	UPOS 116-1,21 Issue154
215	45/ Gesture Control	Summary Properties “My use after “item typo correction	The description of “May use after” items of several properties were changed from “ open ” to “ Not supported ” since it was incorrect information after checking. The properties that this changes apply are AutoDisable DataCount and DataEventEnabled .	UPOS 116-1,21 Issue149
216	45/ Gesture Control	Summary Properties	This device handles the “Hard Totals” device, therefore, CapAssociatedHardTotalsDevice , CapStorage and Storage properties were newly added.	UPOS 116-1,21 Issue144
217	45/ Gesture Control	Summary clearOutput Method	clearOutput method was supported in this device, therefore, “May use after” times are corrected from “ No supported ” to “1.16”.	UPOS 116-1,21 Issue149
218	45/ Gesture Control	Summary getPosition Method	getPosition method parameter value was incorrect, therefore, “ position :int32 by reference ” was corrected to “ out position : int32”.	UPOS 116-1,21 Issue119
219	45/ Gesture Control	Summary Events	TransitionEvent description was added from Ver. 1.16 and this was missing.	UPOS 116-1,21 Issue151

UPOS Ver1.16 RCSD Specification

220	45/ Gesture Control	Model	<p>1. When method make the output for the application. There was a way of data handling. In that section, there was a number in front of each paragraph. However, other section did not use the number therefore those number was eliminated to fit with others. They were as follows.</p> <p>“•1 Buffers the request..... “</p> <p>“•2. Sets the OutputID.....”</p> <p>“•3. Returns as soon as.....”</p> <p>Now they were changed.</p> <p>“• Buffers the request..... “</p> <p>“• Sets the OutputID.....”</p> <p>“• Returns as soon as.....”</p> <p>2. To make precise & real time device control, added the StatusUpdateEvent handling to this device. Then added the description to do so. That is say, “The application will be informed about any status change with a StatusUpdateEvent, also all corresponding status properties will be updated before event delivery.”</p>	UPOS 116-1,21 Issue110 Issue111
221	45/ Gesture Control	Pose/Motion	<p>1. To make a better wording, current description, “Application can then create a pose file by setting the value you want to be defined as a pose with the setPosition method and calling the createPose method.” was changed to, “Then, application can create a pose file by setting the value defined as a pose with the setPosition method and calling the createPose method.”</p> <p>2. Motion file explanation was not good enough. Therefore, description explain the motion file, “A motion file can be created by specifying the pose defined by the created pose file or device and calling the createMotion method.” was changed to, “A motion file can be created and recorded by specifying the pose defined in by the created pose file or the pose defined in the device and creating it as a series of continuously changing actions and calling the createMotion method.”</p> <p>3. HardTotals handling is newly added in this device therefore, current storage function description, “Since the created pause and motion files are recorded in the area managed by the “Hard Totals” service, the application must also support “Hard Totals” service.” was changed to, “Since the created pause and motion files are recorded in the area managed by may store in either the “Hard Totals” service, the application must also support “Hard Totals” service devices or the host file system, or both, and the CapStorage property will show the device’s data file storage location capability.”</p> <p>In addition HardTotals and storage function related description was newly added as follows.</p> <p>“ If device supports either of both Hard Totals devices and the host file system, the application should set the Storage property accordingly to tell where to write the data file.</p> <p>If device needs to be able to write the pose and motion files to a Hard Totals device, the CapAssociatedHardTotalsDevice property holds the open name of the associated Hard Totals device.”</p>	UPOS 116-1,21 Issue112 Issue144
222	45/ Gesture	AutoMode Property	<p>1. In “Remarks” section description of “ If you set one of the properties described in the AutoModeList property</p>	UPOS 116-1,21

UPOS Ver1.16 RCSD Specification

	Control		for this property , the automatic control mode will be enabled in the set mode.” was changed as, “ If you set one of the properties described in the AutoModeList property is set for this property , the automatic control mode will be enabled in the set mode.” for better wording.	Issue115
223	45/ Gesture Control	AutoModeList Property	1. In “Remarks” section, to make the list of joint automatic control IDs precisely, “ and the mode of tracking by moving all joints are supported as follows. ” was eliminated and changed to, “ this is “FaceTrack_Joint01”.” And, “ Another example, in conjunction with the camera, if the mode of tracking the face of a person by moving all joints are supported, this is “FaceTrack_ALL”.” was newly added.	UPOS 116-1,21 Issue115
224	45/ Gesture Control	CapAssociatedHardTotalsDevice Property	This device will handle the “Hard Totals” device, therefore, CapAssociatedHardTotalsDevice property is newly added.	UPOS 116-1,21 Issue144
225	45/ Gesture Control	CapMotion Property	To make the making the motion function precisely, description when it is false and true was changed. Therefore, “ If true, the device supports pose function. If false, the device does not support pose function. ” was changed to, “ If true, the device supports pose making the motion function. Otherwise, it is false. If false, the device does not support pose function. ” And, “ If this property is false, change of PoseCreationMode property, startPose method, createPose method is not available. ” was changed to, “ If When this property is false, change of PoseCreationMode property, startPose method, createPose method is not available startMotion method, createMotion method is not available. ”	UPOS 116-1,21 Issue116
226	45/ Gesture Control	CapPose Property	1. In “Remarks” section, “ If false, the device does not support pose function. ” was eliminated and replaced, “Otherwise, it is false.” for better wording. 2. To make the this pose function precisely, “ If this property is FALSE, change of PoseCreationMode property, startPose method, and createPose method is not available.” was changed to, “ If When this property is FALSE, change of PoseCreationMode property value cannot be changed, in addition, startPose method and createPose method is are not available.”	UPOS 116-1,21 Issue118
227	45/ Gesture Control	CapPoseCreation Property	In “Remarks” section, “ If this property is FALSE, you cannot use the createPose method to change the PoseCreationMode property.” was changed to “ If When this property is FALSE, you cannot use the createPose method that can to change the PoseCreationMode property is not available. ” for better wording.	UPOS 116-1,21 Issue118
228	45/ Gesture Control	CapStorage Property	This device will handle the “Hard Totals” device, therefore, CapStorage property is newly added.	UPOS 116-1,21 Issue144
229	45/ Gesture Control	JointList Property	1. Regarding the JointID description, to make it clear, “ If 0, the joint does not have the position range, 1 holds the position range. For example, the arm joint has a range of rotation width, but the wheel for movement does not have the range of movement amount. ” and “ For example, for a device that supports pitch, roll, and yaw joints and a device that supports rotation by wheel and joint that can move forward and backward, it is as follows. ” were eliminated and replaced as follows, “If position range is 0, the Joint does not have the position range.”	UPOS 116-1,21 Issue114

UPOS Ver1.16 RCSD Specification

			<p>If position range is 1, the joint holds the position range.” and “ For example, arm joint has a range of rotation width but wheel for movement does not have the range of movement amount.</p> <p>If there is a device with joints that supports pitch, roll, yaw and wheels that supports rotating and moving back and forth. In this case they are indicated as follows:”</p>	
230	45/ Gesture Control	MotionList Property	<p>1. To make the description comprehensive added the description regarding the examples as follows. “For example, “bowing, welcoming, clapping...””</p>	UPOS 116-1,21 Issue117
231	45/ Gesture Control	PoseList Property	<p>1. To make the description comprehensive added the description regarding the examples as follows. “For example, “surprise, bow, think...””</p>	UPOS 116-1,21 Issue118
232	45/ Gesture Control	Storage Property	<p>This device will handle the “Hard Totals” device, therefore, Storage property is newly added.</p>	UPOS 116-1,21 Issue144
233	45/ Gesture Control	Table of Gesture Control Device Listed Items in Property	<p>1. To make clear the relationship between property, Item ID, File Name, Name and Parameters, Listed Items Table was newly added.</p>	UPOS 116-1,21
234	45/ Gesture Control	createMotion Method	<p>1. In “Remarks” section, to make clear the relationship between motion file creation and recording, “Specify the registered pose and record it in the motion file.” was eliminated and “A motion file can be created and recorded by specifying the pose defined in the created pose file or the pose defined in the device and creating it as a series of continuously changing actions.”</p> <p>2. Since HardTotals will be used as storage device, the behavior of Storage property was newly added and current description, “The place where the motion file is recorded is the area managed by the “hard totals” device” was changed to “The place where the motion file is recorded is the area managed by the “hard totals” device value of the Storage property.”</p>	UPOS 116-1,21 Issue116
235	45/ Gesture Control	createPose Method	<p>1. In “Remarks” section, to make better wording the current description, “ Before calling this method, you need to set the PoseCreationMode property to TRUE and enable pose registration mode.” Was changed to, “Before calling this method, you it needs to set the PoseCreationMode property to TRUE and to make enableing pose registration mode.”</p> <p>2. Since Storage property is newly added and its description was newly added. Therefore, current description, “The place where the motion file is recorded is the area managed by the “hard totals” device.” was changed to, “The place where the motion file is recorded is the area managed by the “hard totals” device value of the Storage property”</p>	UPOS 116-1,21 Issue118 Issue144
236	45/ Gesture Control	getPosition Method	<p>1. Parameter of this method was changed from “position: int32 by reference” to “out position:int32”</p> <p>2. Regarding the <i>JointID</i> parameter description, “Specify the joint ID” was changed to, “Specify the one of the joint ID values that are listed in the JointList property.” to make clear the between the JointList property. And also, “ Specify one of the values listed in the JointList property. However, it must be an ID whose position range exists or not.” was eliminated and replaced to, “And specified JointList property should be the position range present one.”</p> <p>3. Regarding the position parameter description, current one, “The position of the joint specified by JointID is</p>	UPOS 116-1,21 Issue119

UPOS Ver1.16 RCSD Specification

			<p>stored.” was eliminated and “Store the specified value as the position associated with jointID.” Was newly added.</p> <p>4. In “Remarks” section, “It acquires the position specified by jointID and stores it in position.” was eliminated and “It acquires the position specified by jointID and stores it in position.” was newly added for more clear explanation.</p>	
237	45/ Gesture Control	setPosition Method	<p>1. Regarding the time parameter, its description “Specify the time to control completion” was changed “Specify the time of device to control completion”.</p> <p>2. Regarding the <i>JointID</i> parameter, there was a description, “However, it must be an ID whose position range exists or not.” was corrected as “However, it must be an ID whose position range is present exists or not.”</p> <p>3. In “Remarks” section, there were words of “control” and they were changed to “device control” to fit with the UPOS historical description.</p>	UPOS 116-1,21 Issue152
238	45/ Gesture Control	setSpeed Method	<p>1. In time parameter, there was a word of “control” and changed to “device control” to fit with the UPOS historical description.</p> <p>2. And the description, “If you specify FOREVER(-1), it will continue to operate until you call the stopControl method.” was changed to, “ If you specify the value of FOREVER(-1) is specified, it will continue to operate until you call the stopControl method.” for better wording.</p> <p>3. In “Remarks” section, there was a description of “control” and it was changed to “device control” to fit with the UPOS historical description.</p>	UPOS 116-1,21 Issue122
239	45/ Gesture Control	startMotion Method	<p>1. Regarding the <i>fileName</i> parameter description, “Specify the name of the motion file to start. Or one of the motion ID lists listed in the MotionList property.” was eliminated and “Prior to start this method, need to specify the name of the motion file or the motion ID value that is listed in the MotionList property.” was newly added.</p> <p>2. In “Remarks” section description, “Motion files need to be placed in the area managed by "hard total" service. This method is executed asynchronously. To terminate motion control prematurely, call the stopControl method.” was eliminated and “This method is executed asynchronously and when the device successfully completes a request, an OutputCompleteEvent is enqueued and a property of corresponding event’s OutputID is placed into the OutputID property. The application should compare the returned OutputCompleteEvent property outputID value set by this method to track what data has been sent to device. Motion files are placed in the area as the value of Storage property.” was newly added instead.</p>	UPOS 116-1,21 Issue120
240	45/ Gesture Control	startPose Method	<p>1. There were typo of several “pause” and they were corrected to “pose”.</p> <p>2. There was a description, “ Begin poause defined by the poause file or device specified by fileName.” and typo and some description were changed to make clear description as, “Begin pause-Start the poause defined by the poause file or device specified by fileName.”</p> <p>3. Description after the “This method is executed asynchronously”, “ and when the device successfully completes a request, an OutputCompleteEvent is enqueued and a property of</p>	UPOS 116-1,21 Issue121

UPOS Ver1.16 RCSD Specification

			<p>corresponding event's OutputID is placed into the OutputID property. The application should compare the returned OutputCompleteEvent property OutputID value set by this method to track what data has been sent to device.” was newly added to make precise and real time device control by using the OutputCompleteEvent.</p> <p>4. And “Pose files must be placed in the area managed by "hard total" service.” was eliminated and “ Pose files are placed in the area as the values of Storage property” was newly added since Hard Totals is utilized in this device and related Storage property will be used.</p>	
241	45/ Gesture Control	stopControl Method	<p>1. Regarding the <i>outputID</i> parameter description, “Specify the value of the OutputID property you wish to terminate.” was changed to “Specify the value of the OutputID property to be you wish to terminated.” for better wording.</p> <p>2. In “Remarks” section, “When device successfully complete the request, and OutputCompleteEvent is enqueued. A property of this event contains the <i>outputID</i> of the completed request. The application should compare the returned OutputCompleteEvent property <i>OutputID</i> value with <i>OutputID</i> value set by this method.” was newly added since OutputCompleteEvent will be used for precise device handling.</p>	UPOS 116-1,21 Issue152
242	45/ Gesture Control	DirectIOEvent Event	Since DirectIOEvent related description was missing, therefore DirectIOEvent description was newly added.	UPOS 116-1,21 Issue154
243	45/ Gesture Control	ErrorEvent Event	1.Since ErrorEvent related description was missing, therefore ErrorEvent description was newly added.	UPOS 116-1,21 Issue154
244	45/ Gesture Control	OutputCompleteEvent Event	Since OutputCompleteEvent related description was missing, therefore OutputCompleteEvent description was newly added.	UPOS 116-1,21 Issue154
245	45/ Gesture Control	StatusUpdateEvent Event	Since StatusUpdateEvent related description was missing, therefore StatusUpdateEvent description was newly added.	UPOS 116-1,21 Issue154
246	46/ Device Monitor	Summary Properties	1. Typo was corrected, since OutputID property was described in “May use after” section as “Not Supported”. This was changed to “Not supported.”	UPOS 116-1,21 Issue149
247	46/ Device Monitor	Summary Methods	In the “version” section summary method table, there was an incorrect description regarding the clearInput and clearInputProperties methods Since this method is utilized in this device, it is corrected from “ Not supported ” to “1.16”.	UPOS 116-1,21 Issue149
248	46/ Device Monitor	Summary getDeviceValue method	getDeviceValue method parameter was wrong and it was corrected from “ inout value:int32” to “ pV value:int32”.	UPOS 116-1,21 Issue155
249	46/ Device Monitor	Summary Events	1. Description of “ TransitionEvent ” was added, since it was missing.	UPOS 116-1,21 Issue151
250	46/ Device Monitor	Model	<p>1. Device Monitor measured value handling explanation was not good enough therefore, description, “In the Monitor device control, the measured value of the devices is managed with an integer value of the int32 type, but some devices handle decimal values. In that case, you can calculate the actual value by dividing the measured value by the factor for each device that can be acquired with the DeviceList property.” was changed to “ In the Device Monitor device control, the measured values of the devices isare managed with an integer value most of cases with the int32 type integers,</p>	UPOS 116-1,21 Issue123

UPOS Ver1.16 RCSD Specification

			but some are devices handle decimals values. In that case, the decimals are implicit, you can calculate and the actual value can be calculated by dividing the measured value by the factor for each device the coefficient of each device that can be acquired with obtained in the DeviceList property.” 2. To make precise device handling StatusUpdateEvent will be used, therefore, description, “The application will be informed about any status change with a StatusUpdateEvent, also, all corresponding status properties will be updated before event delivery.” was newly added.	
251	46/ Device Monitor	addMonitoringDevice Method DMON_MMODE_HIGH	1. In the MonitoringMode Value Description, in DMN_MMODE_HIGH, there was a description, “ we will notify the event each time” was changed to “ we will notify the event will be notified in each time” for better wording. 2. Other description “When set to this mode” was changed to “When it is set to this mode”	UPOS 116-1,21 Issue156
252	46/ Device Monitor	addMonitoringDevice Method DMON_MMODE_LOW	1. In the MonitoringMode Value Description, in DMN_MMODE_LOW, there was a description, “ we will notify the event each time” was changed to “ we will notify the event will be notified in each time” for better wording. 2. Other description “When set to this mode” was changed to “When it is set to this mode”	UPOS 116-1,21 Issue156
253	46/ Device Monitor	addMonitoringDevice Method DMON_MMODE_OUTSIDE	1. In the MonitoringMode Value Description, in DMN_MMODE_OUTSIDE, there was a description, “ we will notify the event each time” was changed to “ we will notify the event will be notified in each time” for better wording.	UPOS 116-1,21 Issue156
254	46/ Device Monitor	getDeviceValue Method	1. In this method, there was a parameter of “inout value:int32” this was corrected, “ inout value pValue:int32”. 2. Parameter “value” was changed to “pValuevalue” and also description was changed from “ Measured value obtained from the device. ” to “Pointer that stores measurement values obtained from the device.”	UPOS 116-1,21 Issue155
255	46/ Device Monitor	DataEvent Event	Since DataEvent related description was missing, therefore DataEvent description was newly added.	UPOS 116-1,21 Issue154
256	46/ Device Monitor	DirectIOEvent Event	Since DirectIOEvent related description was missing, therefore DirectIOEvent description was newly added.	UPOS 116-1,21 Issue154
257	46/ Device Monitor	ErrorEvent Event	Since ErrorEvent related description was missing, therefore ErrorEvent description was newly added.	UPOS 116-1,21 Issue154
258	46/ Device Monitor	StatusUpdateEvent Event	Since StatusUpdateEvent related description was missing, therefore StatusUpdateEvent description was newly added.	UPOS 116-1,21 Issue154
259	47/ Graphic Display	Summary Properties Typo	In the “May use after” section summary properties table, there were incorrect description regarding the AutoDisable, DataCount and DataEventEnabled properties. Since those properties were not utilized in this device, they were corrected from “open” to “Not supported”.	UPOS 116-1,21 Issue149
260	47/ Graphic Display	Summary Properties Hard Totals Device	This device handles the “Hard Totals” device, therefore, CapAssociatedHardTotalsDevice, CapStorage and Storage properties were newly added.	UPOS 116-1,21 Issue144
261	47/ Graphic	Summary Properties Add URL wording	There were properties to utilize the URL, however currently it was not included in the URL related property	UPOS 116-1,21

UPOS Ver1.16 RCSD Specification

	Display		names. Therefore, to make the property function clear URL name was added into the CapBack and CapForward properties. Then their names were changed to CapURLBack and CapURLForward .	Issue132
262	47/ Graphic Display	Summay ImageTypeList, VideoTypeList Properties	To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX, XXX and XXXList type of properties. Therefore, CapImageTypeList and CapVideoTypeList property names were changed to ImageTypeList and VideoTypeList . In addition, CapImageType, CapVideoType, ImageType and VideoType properties were newly added.	UPOS 116-1,21 Issue141
263	47/ Graphic Display	Summary Methods Add URL	There were methods utilizing the URL, however currently URL wording was not included in the URL related method names. Therefore, to make the method function clear, URL name was added into the cancelLoading, goBack, goForward and updatePage methods. Then their names were changed to cancelURLLoading, goURLBack, goURLForward, and updateURLPage .	UPOS 116-1,21 Issue136 Issue137 Issue138 Issue139
264	47/ Graphic Display	Summary Events	1. Description of " TransitionEvent " was added, since it was missing.	UPOS 116-1,21 Issue151
265	47/ Graphic Display	Model	In this device, currently there are 3 modes. They are Image display, Movie display and Web display modes. Regarding the Movie display mode to make the harmonization with other devices Movie display mode was changed to MovieVideo display mode.	UPOS 116-1,21 Issue127 Issue129
266	47/ Graphic Display	Image display mode	1. In this mode there was a description, " The application calls the loadImage method to display the image. The CapImageTypeList property lists image files that the device can display. Applications need to support "hard total" services as image files displaying with loadImage method must be placed in the area managed by the "hard total" service. " was eliminated and "The application calls the loadImage method to display the image. The CapImageTypeList property lists image files that the device can display. Applications need to support "hard total" services as image files displaying with loadImage method must be placed in the area managed by the "hard total" service. Prior to start this mode, need to set the appropriate image type file value in the ImageType property from the listed values in the ImageTypeList property, if CapImageType property is true. Then the application can call the loadImage method to display the image. Raises StatusUpdateEvent at the status change timing of image load start with status GDSP_SUE_START_IMAGE_LOAD , and image load end with status GDSP_SUE_END_IMAGE_LOAD . The ImageTypeList property lists image files that the device can display. Applications may need to support "Hard Totals" services as image files displaying with loadImage method might be placed in the area managed by the associated "Hard Totals" service device. If the CapStorage is either GDSP_CST_ALL or GDSP_CST_HARDTOTALS_ONLY , it is possible to store it in the Associated Hard Totals device and storage	UPOS 116-1,21 Issue126

UPOS Ver1.16 RCSD Specification

		<p>device's open name is held in the CapAssociatedHardTotalsDevice property. If device supports both Hard Totals device and the host file system, the application should set the Storage property accordingly to tell where to write the image data file." was newly added to make precise device control by using the StatusUpdateEvent and data storage device.</p>	
<p>267</p>	<p>47/ Graphic Display</p>	<p>MovieVideo Display Mode</p> <p>1. Regarding the graphics control mode explanation was newly added as, "The graphics control of video display modes are as follows." And current description, "The application calls a playVideo method to start playing video." was eliminated. And description, "Prior to start this mode, need to set the appropriate video type file value in the VideoType property from the listed values in the VideoTypeList property, if CapVideoType property is true. Then the application can call the playVideo method to display the video. Also, the video being displayed is stopped by calling the stopVideo method. Raises StatusUpdateEvent at the status change timing of start play video with status GDSP_SUE_START PLAY_VIDEO, and stop play video with status GDSP_SUE_STOP_PLAY_VIDEO." was newly added.</p> <p>2. When device make a validation. After device validation complete successful, then device makes following. In this section, there was a number in front of each paragraph. However, other section did not use the number therefore those number was eliminated to fit with others. They were as follows. "• 1. Buffers the request..... " "• 2. Sets the OutputID....." "• 3. Returns as soon as....." Now they were changed. "• Buffers the request..... " "• Sets the OutputID....." "• Returns as soon as....." 3. Since CapVideoTypeList was changed to VideoTypeList, CapVideoType and VideoType properties. Therefore, VideoDisplayMode related description was changed. That is to say, "The CapVideoTypeList property lists video files that the device can play." was eliminated. And "The video files that the device can display are listed in the VideoTypeList property." was newly added. And "Applications need to support "hard total" services as video files played with the playVideo method must be placed in the area managed by the "hard total" service." was eliminated. And "Since video files to be displayed using the playVideo method must be placed in an area managed by the associated "Hard Totals" service device. If the CapStorage is either GDSP_CST_ALL or GDSP_CST_HARDTOTALS_ONLY, it is possible to store it in the Associated Hard Totals device and storage device's open name is held in the CapAssociatedHardTotalsDevice property. If device supports either or both Hard Totals device and the host file system, the application should set the Storage property accordingly to tell where to write the</p>	<p>UPOS 116-1,21 Issue127</p>

UPOS Ver1.16 RCSD Specification

			<p>image data file.</p> <p>The video display mode of graphics control follows an asynchronous output model. Raises StatusUpdateEvent if Graphic Display device power status or a device status changes are occurred during the video displaying.” was newly added.</p>	
268	47/ Graphic Display	Web Display Mode	<p>To make precise device handling with the StatusUpdateEvent, description was changed a lot and “The web display mode of the Graphics Display follows the general “Device Input Model” for event-driven input: When input is received from the Graphics Display, a DataEvent is enqueued. If the AutoDisable property is true, then the device automatically disables itself when a DataEvent is enqueued. An enqueued DataEvent can be delivered to the application when the DataEventEnabled property is true and other event delivery requirements are met. Just before delivering this event, data is copied into corresponding properties, and further data events are disabled by setting DataEventEnabled to false. This causes subsequent input data to be enqueued while the application processes the current input and associated properties. When the application has finished processing the current input and is ready for more data, it reenables events by setting DataEventEnabled to true. An ErrorEvent (or events) is enqueued if an error occurs while gathering or processing input, and is delivered to the application when DataEventEnabled is true and other event delivery requirements are met. The DataCount property may be read to obtain the total number of enqueued DataEvents. All enqueued input may be deleted by calling clearInput. See the clearInput method description for more details. All data properties that are populated as a result of firing a DataEvent or ErrorEvent can be set back to their default values by calling the clearInputProperties method. The load state of the web page is stored in the LoadStatus property, and the URL is stored in the URL property.” was eliminated. And “The web display mode of graphics control is as follows. The application calls the loadURL method to display the web page. Raises StatusUpdateEvent at the timing of Web page load start with status GDSP_SUE_START_LOAD_WEBPAGE, load finish with status GDSP_SUE_FINISH_LOAD_WEBPAGE, and load cancel with status GDSP_SUE_CANCEL_LOAD_WEBPAGE. And application can detect the web page loading status. The latest loading status of the web page is stored in the LoadStatus property when loadURL method is called, and its URL information is stored in the URL property. In case when cancelLoading method is called during the loading process, current accessed URL information will be stored in the URL property. The graphics control web display mode follows an asynchronous output model.” was newly added.</p>	UPOS 116-1,21 Issue128
269	47/ Graphic Display	CapAssociatedHardTotalsDevice Property	<p>This device will handle the “Hard Totals” device, therefore, CapAssociatedHardTotalsDevice property is newly added.</p>	UPOS 116-1,21 Issue144
270	47/	CapImageType property	<p>To fit with the historical UPOS property handling,</p>	UPOS

UPOS Ver1.16 RCSD Specification

	Graphic Display		CapXXXList properties are changed combination of CapXXX , XXX and XXXList type of properties. Therefore, CapImageType property was newly added.	116-1,21 Issue141
271	47/ Graphic Display	CapStorage Property	This device will handle the “Hard Totals” device, therefore, CapStorage property is newly added.	UPOS 116-1,21 Issue144
272	47/ Graphic Display	CapURLBack Property	1. To make clear URL related property, added the URL word into the property name and CapBack was changed to CapURLBack property. 2. In the “Remarks” section, there was a word of control. To fit with the current UPOS wording, “control” was changed to “ device control ”. 3. In the “Remarks” section there was a description of CapBack property name and it is changed to CapURLBack . 4. In “See also” section there was a method name goBack method this was changed to goURLBack .	UPOS 116-1,21 Issue132
273	47/ Graphic Display	CapURLForward Property	1. To make clear URL related property, added the URL word into the property name and CapForward was changed to CapURLForward property. 2. In the “Remarks” section, there was a word of control. To fit with the current UPOS wording, “control” was changed to “ device control ”. 3. In the “Remarks” section there was a description of CapForward property name and it is changed to CapURLForward . 4. In “See also” section there was a method name goForward method this was changed to goURLForward .	UPOS 116-1,21 Issue132
274	47/ Graphic Display	CapVideoType property	To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX , XXX and XXXList type of properties. Therefore, CapVideoType property was newly added.	UPOS 116-1,21 Issue141
275	47/ Graphic Display	DisplayMode Property	1. Displaying mode Values were incorrect. They were GDISP_DMODE_HIDDEN GDISP_DMODE_IMAGE_FIT GDISP_DMODE_IMAGE_FILL GDISP_DMODE_IMAGE_CENTER GDISP_DMODE_VIDEO_NORMAL GDISP_DMODE_VIDEO_FULL GDISP_DMODE_WEB And Device shortened name was GDISP and this should be GDSP instead. Therefore, those values were changed. GD I SP_DMODE_HIDDEN GD I SP_DMODE_IMAGE_FIT GD I SP_DMODE_IMAGE_FILL GD I SP_DMODE_IMAGE_CENTER GD I SP_DMODE_VIDEO_NORMAL GD I SP_DMODE_VIDEO_FULL GD I SP_DMODE_WEB 2. Regarding GD I SP_DMODE_HIDDEN value meaning, there was a description, “ Hide the screen. ”. It was eliminated and replaced by “ It is a mode to hide images and/or video ” for better wording. 3. Regarding the GD I SP_DMODE_IMAGE_FIT value meaning, there was a description, “the size that maintains the aspect and just enter the screen.” This was changed to “the size that maintains the aspect and just enter fits on ”	UPOS 116-1,21 Issue146

UPOS Ver1.16 RCSD Specification

			<p>the screen.” for better wording.</p> <p>4. Regarding the GDISP_DMODE_VIDEO_NORMAL value meaning, there was a description of “movie” and since this wording is replaced to “movievideo” in this device. Therefore, same wording change should be done.</p> <p>5. In addition, there was a description has a word of movie, “If application hide other modes and screens while displaying images, movies, or web, all displayed contents will be cleared. The movie will be stopped while the movie is playing.” And movie wordings were replaced by vide like this, “If application hide other modes and screens while displaying images, moviesvideos, or web, all displayed contents will be cleared. The movievideo will be stopped while the movievideo is playing.”</p> <p>6. In “See also” section, CapCaptureColorSpaceList, VideoCaptureMode Properties are there as reference and they are Video Capture Device properties. Therefore added the explanation, “” (They are Video Capture Device Properties) and readFrameMethod was eliminated since this method was eliminated its device spec. already.</p>	
276	47/ Graphic Display	ImageType Property	<p>To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX, XXX and XXXList type of properties. Therefore, ImageType property was newly added.</p>	UPOS 116-1,21 Issue141
277	47/ Graphic Display	Cap ImageTypeList Property	<p>1. To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX, XXX and XXXList type of properties. Therefore, CapImageTypeList property name was changed to CapImageTypeList property.</p> <p>2. In “Remarks” section, to explain the relationship ImageType and CapImageType properties, “One of value in the property should be set in the ImageType property, if CapImageType property is true, prior to execute the loadImage method.” was newly added.</p> <p>3. In “See Also” section, CapImageType Property, ImageType Property were newly added as reference.</p>	UPOS 116-1,21 Issue141
278	47/ Graphic Display	LoadStatus Property	<p>1. LoadStatus Property values were incorrect. They were GDISP_LSTATUS_START GDISP_LSTATUS_FINISH GDISP_LSTATUS_CANCEL And Device shortened name was GDISP and this should be GDSP instead. Therefore, those values were changed. GDISP_LSTATUS_START GDISP_LSTATUS_FINISH GDISP_LSTATUS_CANCEL</p> <p>2. Regarding the GDISP_LSTATUS_FINISH meaning value, there was a description, “It have finished loading the web page.” and this was corrected, “It hasve finished loading the web page.”</p> <p>3. Regarding the GDISP_LSTATUS_CANCEL meaning value, there was a description, “It have finished loading the web page.” and this was corrected, “It hasve finished loading the web page.”</p> <p>4. Its value related description was corrected since instead of DataEvent, StatusUpdateEvent will be used. Description, “ Its value is set prior to a DataEvent being delivered to the application.” was changed to, “ Its value</p>	UPOS 116-1,21 Issue131

UPOS Ver1.16 RCSD Specification

			is set prior to a StatusUpdateDataEvent being delivered to the application.”	
279	47/ Graphic Display	Storage Property	To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX, XXX and XXXList type of properties. Therefore, Storage property was newly added.	UPOS 116-1,21 Issue141
280	47/ Graphic Display	URL Property	In “Remarks” its value related description was corrected since instead of DataEvent, StatusUpdateEvent will be used. Description, “ Its value is set prior to a DataEvent being delivered to the application.” was changed to, “ Its value is set prior to a StatusUpdateDataEvent being delivered to the application.”	UPOS 116-1,21 Issue132
281	47/ Graphic Display	VideoType Property	To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX, XXX and XXXList type of properties. Therefore, VideoType property was newly added.	UPOS 116-1,21 Issue141
282	47/ Graphic Display	CapVideoTypeList Property	1. To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX, XXX and XXXList type of properties. Therefore, CapVideoTypeList property name was changed to CapVideoTypeList property. 2. In “Remarks” section, to explain the relationship ImageType and CapImageType properties, “One of value in the property should be set in the VideoType property, if CapImageType property is true, prior to execute the playVideo method.” was newly added. 3. In “See Also” section, CapVideoType Property, VideoType Property were newly added as reference.	UPOS 116-1,21 Issue130 Issue141
283	47/ Graphic Display	cancelURLLoading Method	1. To make clear URL related method, added the URL word into the method name and cancelLoading method was changed to cancelURLLoading method. 2. In the “Remarks” section, there was a description, “The load status is reported by DataEvent or ErrorEvent .” And this was incorrect since instead of DataEvent, StatusUpdateEvent will be used. Therefore, description changed to “The load status is reported by StatusUpdateDataEvent and OutputCompleteEvent or ErrorEvent .”	UPOS 116-1,21 Issue139
284	47/ Graphic Display	goURLBack Method	1. To make clear URL related method, added the URL word into the method name and goBack method was changed to goURLBack method. 2. In the “Remarks” section, there was a description, “The load status is reported by DataEvent or ErrorEvent .” And this was incorrect since instead of DataEvent StatusUpdateEvent will be utilize. Therefore, description changed to “The load status is reported by StatusUpdateDataEvent and OutputCompleteEvent or ErrorEvent .” 3. In “See also” section, CapBack property was changed to CapURLBack property.	UPOS 116-1,21 Issue136
285	47/ Graphic Display	goURLForward Method	1. To make clear URL related method, added the URL word into the method name and goForward method was changed to goURLForward method. 2. In the “Remarks” section, there was a description, “The load status is reported by DataEvent or ErrorEvent .” And this was incorrect since instead of DataEvent StatusUpdateEvent will be used. Therefore, description changed to “The load status is reported by StatusUpdateDataEvent and OutputCompleteEvent or ErrorEvent .”	UPOS 116-1,21 Issue137

UPOS Ver1.16 RCSD Specification

			3. In “See also” section, CapForward property was changed to CapURLForward property.	
286	47/ Graphic Display	loadImage Method	<p>1. Values of Display mode were incorrect. They were They were GDISP_DMODE_IMAGE_FIT GDISP_DMODE_IMAGE_FILL GDISP_DMODE_IMAGE_CENTER And Device shortened name was GDISP and this should be GDSP instead. Therefore, those values were changed. GDISP_DMODE_IMAGE_FIT GDISP_DMODE_IMAGE_FILL GDISP_DMODE_IMAGE_CENTER</p> <p>2. In “Remarks” section, there was a description, “Image files must be located in the area managed by “Hard Totals” service.” And this was changed, “Image files must be are located in the area managed by “Hard Totals” service as the stored values of the Storage property.” since Storage property manage the stored values. And description, “This method is executed asynchronously. Image file loading status is reported by StatusUpdateEvent and OutputCompleteEvent or ErrorEvent.” was newly added.</p>	UPOS 116-1,21 Issue133 Issue146
287	47/ Graphic Display	loadURL Method	<p>1. In “Remarks” section, there was wording “uRL” and it is corrected “uURL”</p> <p>2. In the “Remarks” section, there was a description, “The load status is reported by DataEvent or ErrorEvent.” And this was incorrect since instead of DataEvent StatusUpdateEvent will be used. Therefore, description changed to “The load status is reported by StatusUpdateDataEvent and OutputCompleteEvent or ErrorEvent.”</p>	UPOS 116-1,21 Issue135
288	47/ Graphic Display	playVideo Method	<p>1. In “Remarks” section, to make precise device behavior description, in addition to the current description “Play the specified video”, “that are loaded in the storage area by the loadImage method. All of loaded file images are listed in the ImageTypeList property.” was newly added.</p> <p>2. “Video files must be located in the area managed by “Hard Totals” service.” was changed to, “Video files are must be located in the area managed by “Hard Totals” service as the stored values of the Storage property.” since stored values were handled by Storage property. And, “ The video file playing status will be informed by the StatusUpdateEvent. This method is executed asynchronously. Image file loading status and video file playing status are reported by StatusUpdateEvent and OutputCompleteEvent or ErrorEvent.” since to make precise device handling StatusUpdateEvent will be used instead of DataEvent.</p>	UPOS 116-1,21 Issue134
289	47/ Graphic Display	stopVideo Method	<p>1. In “Remarks” section, instead of DataEvent StatusUpdateEvent and OutputCompleteEvent will be used in this device. Therefore, description, “This method is executed asynchronously. Inage file loading status is reported by StatusUpdateEvent and OutputCompleteEvent or ErrorEvent.” was newly added.</p>	UPOS 116-1,21 Issue157
290	47/ Graphic Display	updateURLPage Method	<p>1. To make clear URL related method, added the URL word into the method name and updatePage method was changed to updateURLPage method.</p> <p>2. To make load status report, it was changed to use</p>	UPOS 116-1,21 Issue138

UPOS Ver1.16 RCSD Specification

			DataEvent to StatusUpdateEvent and OutputCompleteEvent . Therefore, description, “The load status is reported by DataEvent or ErrorEvent .” was changed to, “The load status is reported by StatusUpdateEvent , DataEvent and OutputCompleteEvent or ErrorEvent .”	
291	47/ Graphic Display	DirectIOEvent Event	Since DirectIOEvent related description was missing, therefore DirectIOEvent description was newly added.	UPOS 116-1,21 Issue154
292	47/ Graphic Display	ErrorEvent Event	1.Since ErrorEvent related description was missing, therefore ErrorEvent description was newly added.	UPOS 116-1,21 Issue154
293	47/ Graphic Display	OutputCompleteEvent Event	Since OutputCompleteEvent related description was missing, therefore OutputCompleteEvent description was newly added.	UPOS 116-1,21 Issue154
294	47/ Graphic Display	StatusUpdateEvent Event	Since StatusUpdateEvent related description was missing, therefore StatusUpdateEvent description was newly added.	UPOS 116-1,21 Issue154
295	43/ Sound Player	clearInput Method clearInputProperties Method	clearInput and clearInputProperties Method description was missing.	UPOS 116-1, 21 Issue158
296	44/ Speech Synthesis	clearInput Method clearInputProperties Method	clearInput and clearInputProperties Method description was missing.	UPOS 116-1, 21 Issue158
297	45/ Gestruce Control	clearInput Method clearInputProperties Method	clearInput and clearInputProperties Method description was missing.	UPOS 116-1, 21 Issue158
298	47/ Grphic Display	clearInput Method clearInputProperties Method	clearInput and clearInputProperties Method description was missing.	UPOS 116-1, 21 Issue158
299	47/ Graphic Display	DataEvent	DataEvent will not be used in this device. Therefor, DataEvent description should be eliminated.	UPOS 116-1, 21 Issue145
300	46/ Device Monitor	MonitorigDeviceList Property	The values decribed as follows. “Device01 : 0: 0: 0:0, Device02: 1: 365 :0 :500” This should be corrected, “Device01:0:0:0:0,Devicie02:1:365:0:500” since there is no need the white space.	UPOS 116-1,21 Issue159

UPOS Ver1.16 RCSD Specification

Table 2. Class Diagram Change History Table

Note: If you click the number you can check the actual revised class diagram and can come back here by clicking the Table2-x number in the Class Diagram page.)

No	Chapter/ Device Name	Corrected Items and reason	JIRANo.
1	21/ Lights	“<<property>>FullColor:boolean” was eliminated since it is no need.	UPOS 116-1,21
2	29/ POS Power	<p>1. To make a more precise device battery management in second level, those properties were newly added.</p> <p>“<<capability>>CapBatteryCapacityRemainingInSeconds:boolean” “<<capability>>CapVariableBatteryCrticallyLowThresholdInSeconds:boolean” “<<capability>>CapVariableBatteryLowThresholdInSeconds:boolean” “<<property>>BatteryCapacityRemainingInSeconds:int32” “<<property>>BatteryLowhreshholdInSeconds:int32”</p> <p>2. Battery handling time related properties were newly added and current time managed properties function were already included in the newly added properties. Therefore, there is no need to use the current properties like CapTimeMode and TimeMode properties. Then CapTimeMode and TimeMode properties were eliminated.</p> <p>“<<capability>>CapTimeMode:boolean” “<<property>>TimeMode:boolean”</p> <p>3. POS Power const was added as the status of in second level battery handling as “PWR_SUE_BAT_CAPACITY_REMAINING_In_SECONDS:int32(frozen)” was newly added.</p>	UPOS 116-1,21
3	29 / POS Power	<p>POSPower State Diagram Fig. Chap. 29-4 “POSPoer” typo was corrected to “POSPower”.</p>	UPOS 116-1,21
4	39 /Video Capture	<p>1. DataEvent was eliminated.</p> <p>2. To make property name shorten, wording of “Camera” was eliminated and changed the property name. In addition, exposure and exposition ununified wording was exiting and it was unified as exposure. Consequently current property names are, CapCameraAutoExposition, CapCameraAutoFocus, CapCameraAutoGain, CapCameraAutoWhiteBalance, CapCameraBrightness, CapCameraContrast, CapCameraExposure, CapCameraGain, CapCameraHorizontalFlip, CapCameraHue, CapCameraSaturation, CapCameraVerticalFlip, CameraAutoExposition, CameraAutoFocus, CameraAutoGain, CameraAutoWhiteBalance, CameraBrightness, CameraContrast, CameraExposure, CameraGain, CameraHorizontalFlip, CameraHue, CameraSaturation and CameraVerticalFlip. And they were changed as follows. CapAutoExposure, CapAutoFocus, CapAutoGain, CapAutoWhiteBalance, CapBrightness, CapContrast, CapExposure, CapGain, CapHorizontalFlip, CapHue, CapSaturation, CapVerticalFlip, AutoExposure, AutoFocus, AutoGain, AutoWhiteBalance, Brightness, Contrast, Exposure, Gain, HorizontalFlip, Hue, Saturation and VerticalFlip.</p> <p>They are displayed in the Class diagram as follows. “<<capability>>CapCameraAutoExposure: boolean”, “<<capability>>CapCameraAutoFocus: boolean”, “<<capability>>CapCameraAutoGain: boolean”, “<<capability>>CapCameraAutoWhiteBalance: boolean”, “<<capability>>CapCameraBrightness: boolean”, “<<capability>>CapCameraContrast: boolean”, “<<capability>>CapCameraExposure: boolean”, “<<capability>>CapCameraGain: boolean”, “<<capability>>CapCameraHorizontalFlip: boolean”, “<<capability>>CapCameraHue: boolean”, “<<capability>>CapCameraSaturation: boolean”, “<<capability>>CapCameraVerticalFlip: boolean”,</p>	UPOS 116-1,21

UPOS Ver1.16 RCSD Specification

“<<property>>Auto~~Camera~~Exposure:boolean”,
“<<property>>Auto~~Camera~~Focus:boolean”,
“<<property>>Auto~~Camera~~Gain:boolean”,
“<<property>>Auto~~Camera~~WhiteBalance:boolean”,
“<<property>>~~Camera~~Brightness:int32”,
“<<property>>~~Camera~~Contrast:int32”,
“<<property>>~~Camera~~Exposure:int32”,
“<<property>>~~Camera~~Gain:int32”,
“<<property>>~~Camera~~HorizontalFlip:boolean”,
“<<property>>~~Camera~~Hue:int32”,
“<<property>>~~Camera~~Saturation:int32” and
“<<property>>~~Camera~~VerticalFlip:boolean”

3. To make the property name and method name shorten, wording “photograph” changed to “photo”. Those applied property names are CapPhotograph, CapPhotographType and PhotographType. They were changed as follows. CapPhoto, CapPhotoType and PhotoType properties.

They are displayed in the Class Diagram as follows.
“<<capability>>CapPhoto~~graph~~:boolean”,
“<<capability>>CapPhoto~~graph~~Type:boolean” and
“<<property>>Photo~~graph~~Type:string”.
Also, takePhotograph method was changed to takePhoto method.
Also, takePhoto method’s parameter was changed regarding the overwrite and timeout. Overwrite parameter value was incorrect from int32 to boolean. Timeout parameter value was int32.

It is displayed in the Class Diagram as follows.
“takePhoto~~graph~~(fileName:string, overwrite:~~int32~~boolean, timeout:int32):void”

4. To make the property and method name shorten wording of “Recording” was eliminated.
CapVideoRecordingType and VideoRecordingType were changed to CapVideoType and VideoType. startVideoRecording, stopVideoRecording methods were changed to startVideo and stopVideo methods name.

They are displayed in the Class Diagram as follows.
“+startVideo~~Recording~~(FileName:string, overwrite:boolean, recordingTime:int32):void”
“+stopVideo~~Recording~~():void”

5. Hydra related property names are not needed. Therefore, those properties should not be included. They were BarCodeEnabled, IndividualRecognitionEnabled, CapIndividualRecognition and CapDecodeData properties.

They are not displayed in the Class Diagram as follows.
~~“<<property>>BarCodeEnabled:boolean”,
“<<property>>IndividualRecognitionEnabled:boolean”,
“<<capability>>CapIndividualRecognition:boolean” and
“<<capability>>CapDecodeData:properties:boolean”~~

6. CapCaptureXXX properties are refined to be able to use both Video mode and Photo mode, therefore, functionality of CapCaptureXXX have been ported to both VideoXXX and PhotoXXX properties.
Consequently, newly added properties are CapPhotoColorSpace, CapPhotoFrameRate, PhotoMaxFrameRate, PhotoColorSpaceList, PhotoResolutionList CapVideoColorSpace, VideoMaxFrameRate, VideoColorSpace, VideoColorSpaceList, VideoResolutionList, PhotoTypeList and VideoTypeList.
They are displaying in the class diagram as follows.

UPOS Ver1.16 RCSD Specification

<p> <code>“<<capability>>CapPhotoColorSpace:boolean”,</code> <code>“<<capability>>CapPhotoFrameRate:boolean”,</code> <code>“<<property>>PhotoMaxFrameRate:int32”,</code> <code>“<<property>>PhotoColorSpaceList:string”,</code> <code>“<<property>>PhotoResolutionList:string”</code> <code>“<<capability>> CapVideoColorSpace:boolean”,</code> <code>“<<property>>VideoMaxFrameRate:int32”,</code> <code>“<<property>>VideoColorSpace:string”,</code> <code>“<<property>>VideoColorSpaceList:string”,</code> <code>“<<property>>VideoResolutionList:string”,</code> <code>“<<property>>PhotoTypeList:string”</code> and <code>“<<property>>VideoTypeList:string”</code>. </p> <p> Eliminated properties are CapCaptureFrameRate, CapCaptureMaxFrameRate, CapCapture, CapCaptureColorSpace, CapCaptureColorSpaceList, CapCaptureResolution, CapCaptureResolutionList, CapVideoRecordingMaxFrameRate and CapVideoRecordingResolutionList. </p> <p> They are eliminated in the Class Diagram as follows. </p> <p> “<<capability>>CapCaptureFrameRate:boolean”, “<<capability>>CapCaptureMaxFrameRate:inboolean”, “<<capability>>CapCapture:boolean”, “<<capability>>CapCaptureColorSpace:boolean”, “<<capability>>CapCaptureColorSpaceList:string”, “<<capability>>CapCaptureResolution:boolean”, “<<capability>>CapCaptureResolutionList:string”, “<<capability>>CapVideoRecordingMaxFrameRate:int32”, “<<capability>>CapVideoRecordingResolutionList:string” </p> <p> Those properties, CapPhotographResolution, CaptureColorSpace, CaptureFrameRate, CaptureResolution, CapVideoRecordingFrameRate, CapVideoRecordingResolution, VideoRecordingFrameRate and VideoRecordingResolution were changed to CapPhotoResolution, PhotoColorSpace, PhotoFrameRate, PhotoResolution, CapVideoFrameRate, CapVideoResolution, VideoFrameRate and VideoResolution. </p> <p> They are displayed in the Class Diagrams as follows. </p> <p> “<<capability>>CapPhotographResolution:boolean”, “<<property>>CapturePhotoColorSpace:string”, “<<property>>CapturePhotoFrameRate:int32”, “<<property>>CapturePhotoResolution:string”, “<<capability>>CapVideoRecordingFrameRate:boolean”, “<<capability>>CapVideoRecordingResolution:boolean”, “<<property>>VideoRecordingFrameRate:int32” and “<<property>>VideoRecordingResolution:string”. </p> <p> 7. To fit with the historical UPOS property handling, CapXXXXList properties are changed combination of CapXXX, XXX and XXXList type of properties. Consequently, those properties were eliminated. They are CapPhotographResolutionList, CapVideoRecordingResolutionList, CapPhotographTypeList, CapCaptureResolutionList and CapCaptureColorSpaceList. </p> <p> They are eliminated from the Class Diagram as follows.. </p> <p> “<<capability>>CapPhotographResolutionList:string”, “<<capability>>CapVideoRecordingResolutionList:string”, “<<capability>>CapPhotographTypeList:string”, “<<capability>>CapCaptureResolutionList:string”, “<<capability>>CapCaptureColorSpaceList:string” </p> <p> 8. To handle the video recording precisely, added the new property. It is RemainingRecordingTimeInSec property. It is displayed in the Class Diagram as follows. </p>
--

UPOS Ver1.16 RCSD Specification

		<p>“<<property>>RemainingRecordingTimeInSec:int32”.</p> <p>9. As Video Capture Device specific method, readFrame method was eliminated. It is eliminated from the Class Diagram as follows.</p> <p>“+readFrame(sequenceNumber:int32, type:int32, fileName:string, overwrite:boolean, recordingTime:time out :int32):void”</p>	
<u>5</u>	42 /Sound Recorder	<p>1. This device was handling the “Hard Totals” device, therefore, CapAssociatedHardTotalsDevice, CapStorage and Storage properties are newly added. They are displayed in the Class Diagram as follows.</p> <p>“<<capability>>CapAssociatedHardTotalsDevice:boolean”, “<<capability>>CapStorage:boolean”, “<<property>>Storage:int32”</p> <p>2. To make the precise recording time control RemainingRecordingTimeInSec property was newly added. It is displayed in the Class Diagram as follows.</p> <p>“<<property>>RemaingRecordingTimeInSec:int32”</p> <p>3. To make a Sound Data Handling SoundData property was newly added. It is displayed in the Class Diagram as follows.</p> <p>“<<property>>SoundData:binary”</p> <p>4.To fit with the historical UPOS property handling, CapXXXList properties are changed combination of CapXXX, XXX and XXXList type of properties. Therefore, CapChannelList, CapSamplingRateList and CapSoundTypeList properties were eliminated. And ChannelList, SamplingRateList and SoundTypeList properties were newly added. They are eliminated from the Class Diagram as follows.</p> <p>“<<capability>>CapChannelList:string”, “<<capability>>CapSamplingRateList:string”, “<<capability>>CapSoundTypeList:string”</p> <p>They are displayed in the Class Diagram as follows.</p> <p>“<<property>>ChannelList:string”, “<<property>>SamplingRateList :string”, “<<property>>SoundTypeList:string”.</p>	UPOS 116-1,21
<u>6</u>	43 / Sound Player	<p>1. This device was handling the “Hard Totals” device, therefore, CapAssociatedHardTotalsDevice, CapStorage and Storage properties are newly added. They are displayed in the Class Diagram as follows.</p> <p>“<<capability>>CapAssociatedHardTotalsDevice:boolean”, “<<capability>>CapStorage:boolean”, “<<property>>Storage:int32”</p>	UPOS 116-1,21
<u>7</u>	45 / Gesture Control	<p>1. This device was handling the “Hard Totals” device, therefore, CapAssociatedHardTotalsDevice, CapStorage and Storage properties are newly added. They are displayed in the Class Diagram as follows.</p> <p>“<<capability>>CapAssociatedHardTotalsDevice:boolean”, “<<capability>>CapStorage:boolean”, “<<property>>Storage:int32”</p> <p>2. getPosition method parameter was corrected. Therefore currently it was as follows.</p> <p>“+getPosition (jointID: string, position: int32 by reference): void { raises-exception, use after open, claim, enable }”</p> <p>This was changed: “+getPosition (jointID: string, out position: int32): void { raises-exception, use after open, claim, enable }”</p>	UPOS 116-1,21
<u>8</u>	46 / Device Monitor	<p>1. getDeviceValue “inout value:int32” was changed to pValue:int32. Therefore, this method was changed and displayed in Class Diagram as follows.</p> <p>“+getDeviceValue(deviceID:string, inout valuepValue:int32)</p>	UPOS 116-1,21
<u>9</u>	47 / Graphic Display	<p>1. DataEvent was eliminated, since it is not used.</p> <p>2. This device was handling the “Hard Totals” device, therefore, CapAssociatedHardTotalsDevice, CapStorage and Storage properties are newly added. They are displayed in the Class Diagram as follows.</p> <p>“<<capability>>CapAssociatedHardTotalsDevice:boolean”, “<<capability>>CapStorage:boolean”, “<<property>>Storage:int32”</p> <p>3. To fit with the historical UPOS property handling, CapXXXList properties are</p>	UPOS 116-1,21

UPOS Ver1.16 RCSD Specification

	<p>changed combination of CapXXX, XXX and XXXList type of properties. Therefore, CapImageTypeList and CapVideoTypeList properties were changed to CapImageTypeList and CapVideoTypeList properties. In addition, CapImageType, ImageType, CapVideoType and VideoType properties were newly added. Changed properties are displayed in the Class Diagram as follows. “<<capabilityproperty>>CapImageTypeList:string” “<<capabilityproperty>>CapVideoTypeList:string” Newly added properties are displayed in the Class Diagram as follows. “<<capability>>CapImageType:boolean” “<<property>>ImageType:string” “<<capability>>CapVideoType:beelean” “<<property>>VodepType:string”</p> <p>4. To make clear the property function, add the URL related properties URL names. Currently there were properties named CapBack and CapForward. Those property names were changed CapURLBack and CapURLForward. They will be displayed in Class Diagram as follows. “<<capability>>CapURLBack:boolean” “<<capability>>CapURLForward:boolean”</p> <p>5. To make clear the method function added the URL related methods URL names. Currently there were methods named cancelLoading, goBack, goForward and updatePage. These methods names were changed cancelURLLoading, goURLBack, goURLForward and updateURLPage. Those were displayed in the Class Diagram as follows. “+cancelURLLoading():void” “+goURLBack():void” “+goURLForward():void” “+updateURLPage():void”</p>	
--	--	--

Lights

This Chapter defines the Lights device category.

Summary

Properties (UML attributes)

<i>Common</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
AutoDisable:	<i>boolean</i>	{read-write}	1.12	Not sSupported
CapCompareFirmwareVersion:	<i>boolean</i>	{read-only}	1.12	open
CapPowerReporting:	<i>int32</i>	{read-only}	1.12	open
CapStatisticsReporting:	<i>boolean</i>	{read-only}	1.12	open
CapUpdateFirmware:	<i>boolean</i>	{read-only}	1.12	open
CapUpdateStatistics:	<i>boolean</i>	{read-only}	1.12	open
CheckHealthText:	<i>string</i>	{read-only}	1.12	open
Claimed:	<i>boolean</i>	{read-only}	1.12	open
DataCount:	<i>int32</i>	{read-only}	1.12	Not sSupported
DataEventEnabled:	<i>boolean</i>	{read-write}	1.12	Not sSupported
DeviceEnabled:	<i>boolean</i>	{read-write}	1.12	open & claim
FreezeEvents:	<i>boolean</i>	{read-write}	1.12	open
OutputID:	<i>int32</i>	{read-only}	1.12	Not sSupported
PowerNotify:	<i>int32</i>	{read-write}	1.12	open
PowerState:	<i>int32</i>	{read-only}	1.12	open
State:	<i>int32</i>	{read-only}	1.12	--
DeviceControlDescription:	<i>string</i>	{read-only}	1.12	--
DeviceControlVersion:	<i>int32</i>	{read-only}	1.12	--
DeviceServiceDescription:	<i>string</i>	{read-only}	1.12	open
DeviceServiceVersion:	<i>int32</i>	{read-only}	1.12	open
PhysicalDeviceDescription:	<i>string</i>	{read-only}	1.12	open
PhysicalDeviceName:	<i>string</i>	{read-only}	1.12	open

[Goto Table 1-7](#)
[Goto Table1-8](#)

UPOS Ver1.16 RCSD Specification

Properties (Continued)

<i>Specific</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
CapAlarm:	int32	{read-only}	1.12	open
CapBlink:	boolean	{read-only}	1.12	open
CapColor:	int32	{read-only}	1.12	open
CapFullColor	boolean	{read-only}	1.16	open
CapPattern:	int32	{read-only}	1.16	open
FullColor	boolean	{read-only}	1.16	open
MaxLights:	int32	{read-only}	1.12	open

Methods (UML operations)

Common

<i>Name</i>	<i>Version</i>
open (logicalDeviceName: string): void {raises-exception}	1.12
close (): void {raises-exception, use after open}	1.12
claim (timeout: int32): void {raises-exception, use after open}	1.12
release (): void {raises-exception, use after open, claim}	1.12
checkHealth (level: int32): void {raises-exception, use after open, enable}	1.12
clearInput (): void { }	<i>Not supported</i>
clearInputProperties (): void { }	<i>Not supported</i>
clearOutput (): void { }	<i>Not supported</i>
directIO (command: int32, inout data: int32, inout obj: object): void {raises-exception, use after open}	1.12
compareFirmwareVersion (firmwareFileName: string, out result: int32): void {raises-exception, use after open, enable}	1.12
resetStatistics (statisticsBuffer: string): void {raises-exception, use after open, enable}	1.12
retrieveStatistics (inout statisticsBuffer: string): void {raises-exception, use after open, enable}	1.12
updateFirmware (firmwareFileName: string): void {raises-exception, use after open, enable}	1.12
updateStatistics (statisticsBuffer: string): void {raises-exception, use after open, enable}	1.12

[Goto Teable1-6](#)
[Goto Table 1-9](#)

UPOS Ver1.16 RCSD Specification

Specific

Name

switchOff (lightNumber: int32): void {raises-exception, use after open, claim, enable}	1.12
switchOn (lightNumber: int32, blinkOnCycle: int32, blinkOffCycle: int32, color: int32, alarm: int32): void {raises-exception, use after open, claim, enable}	1.12+16
switchOnMultiple (lightNumbers: string, blinkOnCycle: int32, blinkOffCycle: int32, color: int32, alarm: int32): void {raises-exception, use after open, claim, enable}	1.16
switchOnPattern (pattern: int32, alarm: int32): void {raises-exception, use after open, claim, enable}	1.16
switchOffPattern (): void {raises-exception, use after open, claim, enable}	1.16

Events (UML interfaces)

<i>Name</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>
upos::events::DataEvent		Not sSupported	
upos::events::DirectIOEvent			1.12
EventNumber:	int32	{read-only}	
Data:	int32	{read-write}	
Obj:	object	{read-write}	
upos::events::ErrorEvent		Not sSupported	
upos::events::OutputCompleteEvent		Not sSupported	
upos::events::StatusUpdateEvent			1.12
Status:	int32	{read-only}	
upos::events::TransitionEvent		Not supported	1.16

[Goto Table1-7](#)
[Goto Table1-10](#)

UPOS Ver1.16 RCSD Specification

General Information

The Lights programmatic name is “Lights”.

This device category was added to Version 1.12 of the specification.

Capabilities

- The Lights device control has the following capability:
 - Supports commands to “switch on” and “switch off” a light.
- The Lights device control may have the following additional capabilities:
 - Supports device-level blinking at adjustable blink cycles.
 - Support multiple lights.
 - Supports different colors of a light.
 - Supports different alarms

Device Sharing

Lights is an exclusive-use device. Its device sharing rules are:

- The application must claim the device before enabling it.
- The application must claim and enable the device before accessing some of the properties and methods, or receiving events.
- See the “Summary” table for precise usage prerequisites.

The following diagram shows the relationships between the Lights classes

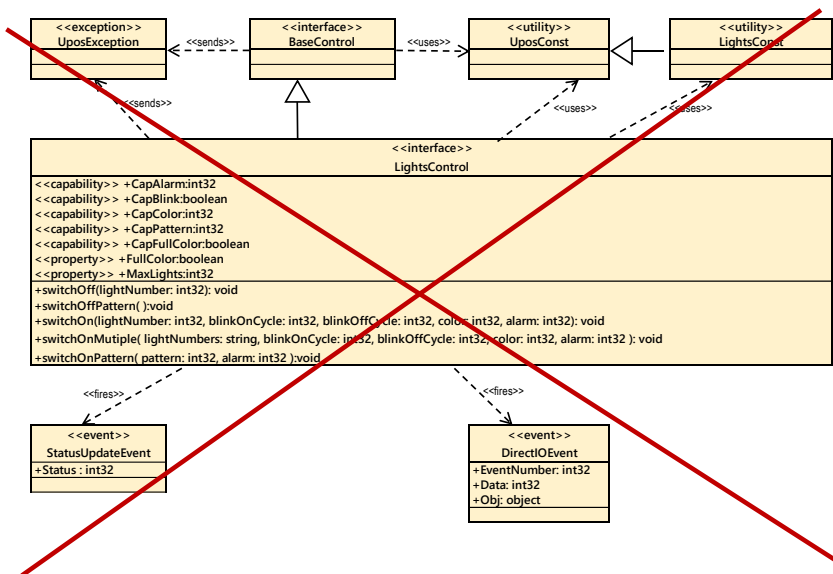
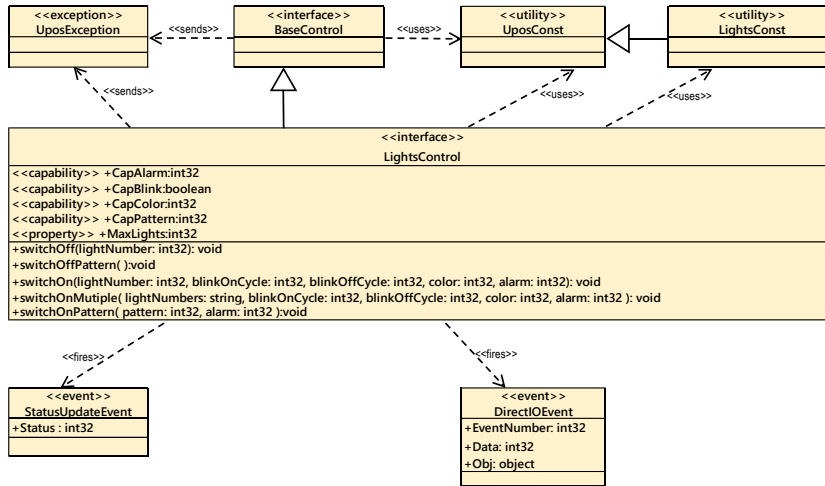


Fig. Chap. 21-1 Lights Class Diagram

[Goto Table 2-1](#)

UPOS Ver1.16 RCSD Specification Lights Sequence Diagram

The following sequence diagram show the typical usage of the Lights device illustrating the handling of the media entry indicator lights.

NOTE : We are assuming that the Application has already successfully opened and claimed the Light Device. MaxLights is 4 defining a SelfCheckout Media Entry Indicator (light1 is BillAcceptor, light2 is BillDispenser, light3 is CoinAcceptor, light4 is CoinDispenser) and that CapBlink is true.

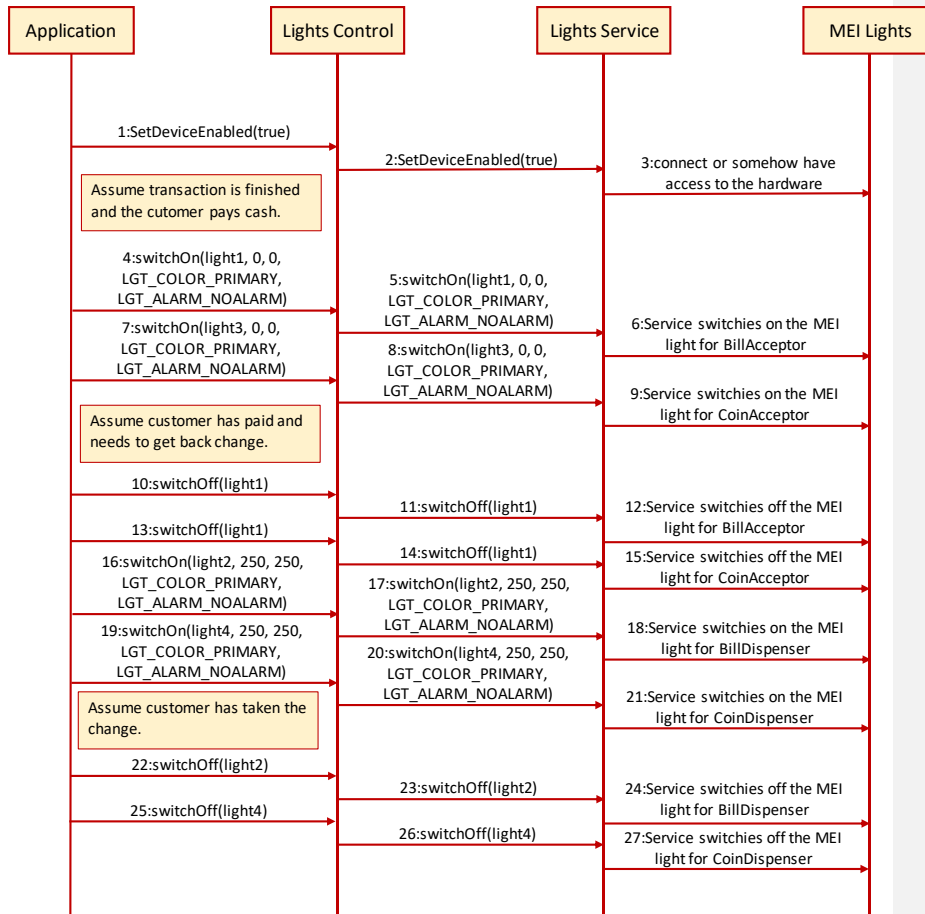


Fig. Chap. 21-2 Lights Sequence Diagram (handling of the media entry indicator lights)

UPOS Ver1.16 RCSD Specification

The following sequence diagram show the typical usage of the Lights device illustrating the handling of the pole lights.

NOTE : We are assuming that the Application has already successfully opened and claimed the Light Device. MaxLights is 3 defining a SelfCheckout Media Entry Indicator (light1 is green, light2 is yellow, light3 is red) and that the device supports alarms.

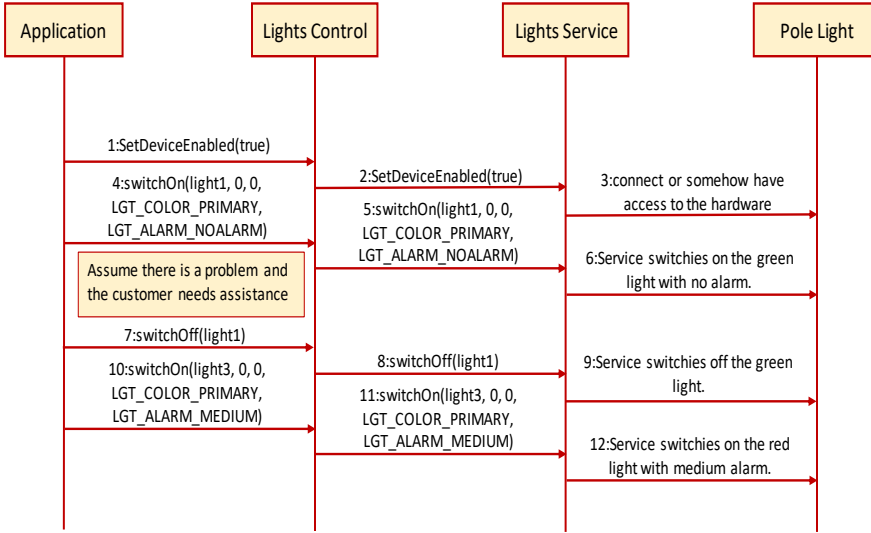


Fig. Chap. 21-3 Lights Sequence Diagram (handling of the pole lights)

UPOS Ver1.16 RCSD Specification

Properties(UML attributes)

CapAlarm Property

Syntax	CapAlarm: <i>int32</i> {read-only, access after open}														
Remarks	<p>This capability indicates if the device supports different alarms.</p> <p>CapAlarm is a logical OR combination of any of the following values:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>LGT_ALARM_NOALARM</td> <td>Alarms are not supported.</td> </tr> <tr> <td>LGT_ALARM_SLOW</td> <td>Supports a slow beep.</td> </tr> <tr> <td>LGT_ALARM_MEDIUM</td> <td>Supports a medium beep.</td> </tr> <tr> <td>LGT_ALARM_FAST</td> <td>Supports a fast beep.</td> </tr> <tr> <td>LGT_ALARM_CUSTOM1</td> <td>Supports 1st custom alarm.</td> </tr> <tr> <td>LGT_ALARM_CUSTOM2</td> <td>Supports 2nd custom alarm.</td> </tr> </tbody> </table> <p>This property is initialized by the open method. If the device does not support alarms, it is initialized to LGT_ALARM_NOALARM.</p>	Value	Meaning	LGT_ALARM_NOALARM	Alarms are not supported.	LGT_ALARM_SLOW	Supports a slow beep.	LGT_ALARM_MEDIUM	Supports a medium beep.	LGT_ALARM_FAST	Supports a fast beep.	LGT_ALARM_CUSTOM1	Supports 1st custom alarm.	LGT_ALARM_CUSTOM2	Supports 2nd custom alarm.
Value	Meaning														
LGT_ALARM_NOALARM	Alarms are not supported.														
LGT_ALARM_SLOW	Supports a slow beep.														
LGT_ALARM_MEDIUM	Supports a medium beep.														
LGT_ALARM_FAST	Supports a fast beep.														
LGT_ALARM_CUSTOM1	Supports 1st custom alarm.														
LGT_ALARM_CUSTOM2	Supports 2nd custom alarm.														
Errors	A UpoException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.														

CapBlink Property

Syntax	CapBlink: <i>boolean</i> {read-only, access after open}
Remarks	<p>If true, a blinking capability is supported. It may be either a physical capability of the device or emulated by the service.</p> <p>This property is initialized by the open method.</p>
Errors	A UpoException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.

CapColor Property

Syntax	CapColor: <i>int32</i> {read-only, access after open}														
Remarks	<p>This capability indicates if the device supports different colors.</p> <p>CapColor is a logical OR combination of any of the following values:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>LGT_COLOR_PRIMARY</td> <td>Supports Primary Color (Usually Green).</td> </tr> <tr> <td>LGT_COLOR_CUSTOM1</td> <td>Supports 1st Custom Color (Usually Red).</td> </tr> <tr> <td>LGT_COLOR_CUSTOM2</td> <td>Supports 2nd Custom Color (Usually Yellow).</td> </tr> <tr> <td>LGT_COLOR_CUSTOM3</td> <td>Supports 3rd Custom Color.</td> </tr> <tr> <td>LGT_COLOR_CUSTOM4</td> <td>Supports 4th Custom Color.</td> </tr> <tr> <td>LGT_COLOR_CUSTOM5</td> <td>Supports 5th Custom Color.</td> </tr> </tbody> </table> <p>This property is initialized by the open method. If the device supports only one color, it is initialized to LGT_COLOR_PRIMARY.</p>	Value	Meaning	LGT_COLOR_PRIMARY	Supports Primary Color (Usually Green).	LGT_COLOR_CUSTOM1	Supports 1st Custom Color (Usually Red).	LGT_COLOR_CUSTOM2	Supports 2nd Custom Color (Usually Yellow).	LGT_COLOR_CUSTOM3	Supports 3rd Custom Color.	LGT_COLOR_CUSTOM4	Supports 4th Custom Color.	LGT_COLOR_CUSTOM5	Supports 5th Custom Color.
Value	Meaning														
LGT_COLOR_PRIMARY	Supports Primary Color (Usually Green).														
LGT_COLOR_CUSTOM1	Supports 1st Custom Color (Usually Red).														
LGT_COLOR_CUSTOM2	Supports 2nd Custom Color (Usually Yellow).														
LGT_COLOR_CUSTOM3	Supports 3rd Custom Color.														
LGT_COLOR_CUSTOM4	Supports 4th Custom Color.														
LGT_COLOR_CUSTOM5	Supports 5th Custom Color.														
Errors	A UpoException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.														

~~CapFullColor Property~~

Added in Release 1.16

Syntax	CapColor: <i>boolean</i> {read-only, access after open}
Remarks	<p>If true, the application can set FullColor property to true and specify full color.</p> <p>If false, the application cannot specify full color.</p> <p>This property is initialized by the open method.</p>
Errors	A UpoException may be thrown when this property is accessed. For further information, see “Errors” on page Intro-20.
See Also	FullColor Property, switchOn Method, switchOnMultiple Method.

[Goto Table 1-11](#)

UPOS Ver1.16 RCSD Specification

CapPattern Property **Added in Release 1.16**

Syntax	CapPattern: int32 {read-only, access after open}						
Remarks	This capability indicates if the device supports different lighting patterns. CapPattern is a logical OR combination of any of the following values: <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>LGT_PATTERN_NOPATTERN</td><td>Lighting patterns are not supported.</td></tr><tr><td>LGT_PATTERN_CUSTOM</td><td>1~32 Supports 1st to 32th Lighting Pattern.</td></tr></tbody></table> This property is initialized by the open method. If the device does not support lighting pattern, it is initialized to LGT_PATTERN_NOPATTERN.	<u>Value</u>	<u>Meaning</u>	LGT_PATTERN_NOPATTERN	Lighting patterns are not supported.	LGT_PATTERN_CUSTOM	1~32 Supports 1 st to 32 th Lighting Pattern.
<u>Value</u>	<u>Meaning</u>						
LGT_PATTERN_NOPATTERN	Lighting patterns are not supported.						
LGT_PATTERN_CUSTOM	1~32 Supports 1 st to 32 th Lighting Pattern.						
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.						
See Also	switchOnPattern Method.						

~~FullColor Property **Added in Release 1.16**~~

Syntax	FullColor: boolean {read-write, access after open}
Remarks	Holds the format of the value to specify for the <i>Color</i> parameter of SwitchOn method and SwitchOnMultiple method. If true, the <i>Color</i> parameter format is full color of 0xRRGGBB format. If false, the <i>Color</i> parameter format is one of the colors defined by CapColor. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “Errors” on page Intro-20.
See Also	CapFullColor Property, switchOn Method, switchOnMultiple Method.

[Goto Table 1-12](#)

MaxLights Property

Syntax	MaxLights: int32 {read-only, access after open}
Remarks	MaxLights specifies the maximum number of lights that the device can support. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.

UPOS Ver1.16 RCSD Specification

Methods (UML operations)

switchOff Method

Syntax **switchOff (lightNumber: int32):**
 void {raises-exception, use after open-claim-enable}

<u>Parameter</u>	<u>Description</u>
<i>lightNumber</i>	Specifies the light number. Valid light numbers are 1 through MaxLights .

Remarks Switches off the light specified by *lightNumber*.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.

A possible value of the exception’s *ErrorCode* property is:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	The <i>lightNumber</i> parameter exceeds MaxLights .

See Also **MaxLights** Property.

switchOffPattern Method

Syntax **switchOff Pattern ():**
 void {raises-exception, use after open-claim-enable}

Remarks Switches off the pattern lighting.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.

A possible value of the exception’s *ErrorCode* property is:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	Pattern lighting is not executed.

See Also **switchOnPattern** Method.

UPOS Ver1.16 RCSD Specification
switchOn Method

Updated in Release 1.12-1.16

Syntax **switchOn** (**lightNumber**: *int32*, **blinkOnCycle**: *int32*,
blinkOffCycle: *int32*, **color**: *int32*, **alarm**: *int32*):
void {raises-exception, use after open-claim-enable}

Parameter	Description
<i>lightNumber</i>	Specifies the light number. Valid light numbers are 1 through MaxLights .
<i>blinkOnCycle</i>	A zero (0) value indicates no blink cycle. A positive value indicates the blink on cycle time in milliseconds. Negative values are not allowed.
<i>blinkOffCycle</i>	A zero (0) value indicates no blink cycle. A positive value indicates the blink off cycle time in milliseconds. Negative values are not allowed.
<i>color</i>	If FullColor is true, specifies the color of the light, must be full color of 0xRRGGBB format. If FullColor is false, specifies the color of the light, must be one of the colors defined by CapColor .
<i>alarm</i>	Specifies the used alarm type, must be one of the alarms defined by CapAlarm .

Remarks Switches on the light specified by *lightNumber* or let it blink.
If *blinkOnCycle* and *blinkOffCycle* are zero (0) or **CapBlink** is false, then the parameters *blinkOnCycle* and *blinkOffCycle* will be ignored and the light will only be switched on.
If **CapBlink** is true and *blinkOnCycle* and *blinkOffCycle* are positive, then the light will blink.
If **CapColor** is LGT_COLOR_PRIMARY the light does not support different colors and *color* is ignored, otherwise **switchOn** will use the color specified by *color*.
If **CapAlarm** is LGT_ALARM_NOALARM the light does not support different alarms and *alarm* is ignored, otherwise **switchOn** will use the alarm specified by *alarm*.
Subsequent calls to **switchOn** will change the blink cycles, the color or the alarm type of the light.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.
A possible value of the exception’s *ErrorCode* property is:

Value	Meaning
E_ILLEGAL	The <i>lightNumber</i> parameter exceeds MaxLights , an invalid <i>color</i> or <i>alarm</i> was specified.

See Also **CapAlarm** Property, **CapBlink** Property, **CapColor** Property, ~~**FullColor** Property~~, **MaxLights** Property.

[Goto Table1-13](#)

switchOnMultiple Method **Added in Release 1.16**

Syntax `switchOnMultiple (lightNumbers: string, blinkOnCycle: int32, blinkOffCycle: int32, color: int32, alarm: int32): void {raises-exception, use after open-claim-enable}`

Parameter	Description
<i>lightNumbers</i>	Specifies the comma-delimited list of light number. Valid light numbers are 1 through MaxLights .
<i>blinkOnCycle</i>	A zero (0) value indicates no blink cycle. A positive value indicates the blink on cycle time in milliseconds. Negative values are not allowed.
<i>blinkOffCycle</i>	A zero (0) value indicates no blink cycle. A positive value indicates the blink off cycle time in milliseconds. Negative values are not allowed.
<i>color</i>	If FullColor is true, specifies the color of the light, must be full color of 0xRRGGBB format. If FullColor is false, specifies the color of the light, must be one of the colors defined by CapColor .
<i>alarm</i>	Specifies the used alarm type, must be one of the alarms defined by CapAlarm .

Remarks This method does the same as swithcOn but in a synchronized way such that all lights are switched on / blinking synchronously. Switches on the multiple lights specified by *lightNumbers* or let it blink.

If *blinkOnCycle* and *blinkOffCycle* are zero (0) or **CapBlink** is false, then the parameters *blinkOnCycle* and *blinkOffCycle* will be ignored and the light will only be switched on.

If **CapBlink** is true and *blinkOnCycle* and *blinkOffCycle* are positive, then the light will blink.

If **CapColor** is LGT_COLOR_PRIMARY the light does not support different colors and *color* is ignored, otherwise **switchOnMultiple** will use the color specified by *color*.

If **CapAlarm** is LGT_ALARM_NOALARM the light does not support different alarms and *alarm* is ignored, otherwise **switchOnMultiple** will use the alarm specified by *alarm*.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.

A possible value of the exception’s *ErrorCode* property is:

Value	Meaning
E_ILLEGAL	The <i>lightNumbers</i> parameter exceeds MaxLights , an invalid value was specified.

See Also **CapAlarm** Property, **CapBlink** Property, **CapColor** Property, ~~**FullColor** Property~~, **MaxLights** Property.

[Goto Table 1-14](#)

UPOS Ver1.16 RCSD Specification
switchOnPattern Method

Added in Release 1.16

Syntax **switchOnPattern** (*pattern: int32, alarm: int32*):
 void {raises-exception, use after open-claim-enable}

Parameter	Description
<i>pattern</i>	Specifies the lighting pattern, must be one of the patterns defined by CapPattern .
<i>alarm</i>	Specifies the used alarm type, must be one of the alarms defined by CapAlarm .

Remarks Switches on the light specified by *pattern*.
If **CapAlarm** is LGT_ALARM_NOALARM the light does not support different alarms and *alarm* is ignored, otherwise **switchOn** and **switchOnPattern** will use the alarm specified by *alarm*.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.

A possible value of the exception’s *ErrorCode* property is:

Value	Meaning
E_ILLEGAL	An invalid value was specified, or unsupported operation with the Device.

See Also **CapAlarm** Property, **CapPattern** Property.

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

DirectIOEvent

<<event>> **upos::events::DirectIOEvent**
EventNumber : *int32* {read-only}
Data : *int32* {read-write}
Obj : *object*{read-write}

Description Provides Service information directly to the application. This event provides a means for a vendor-specific Lights Service to provide events to the application that are not otherwise supported by the device control.

Attributes This event contains the following attributes:

Attribute	Type	Description
<i>EventNumber</i>	<i>int32</i>	Event number whose specific values are assigned by the Service.
<i>Data</i>	<i>int32</i>	Additional numeric data. Specific values vary by the <i>EventNumber</i> and the Service. This property is settable.
<i>Obj</i>	<i>Object</i>	Additional data whose usage varies by the <i>EventNumber</i> and Service. This property is settable.

Remarks This event is to be used only for those types of vendor specific functions that are not otherwise described. Use of this event may restrict the application program from being used with other vendor's Lights devices which may not have any knowledge of the Service's need for this event.

See Also "Events" on page Intro-19, **directIO** Method.

StatusUpdateEvent

<<event>> **upos::events::StatusUpdateEvent**
Status : *int32* {read-only}

Description Notifies the application that there is a change in the power status of a light.

Attributes This event contains the following attribute:

Attribute	Type	Description
<i>Status</i>	<i>int32</i>	Reports a change in the power status of a light. <i>Note that Release 1.3</i> added Power State Reporting with additional <i>Power reporting StatusUpdateEvent values</i> . The Update Firmware capability, added in <i>Release 1.9</i> , added additional <i>Status</i> values for communicating the status/progress of an asynchronous update firmware process. See " StatusUpdateEvent " description on page 1-34.

Remarks Enqueued when the light detects a power state change.

See Also "Events" on page Intro-19.

POS Power

This Chapter defines the POS Power device category.

Summary

Properties (UML attributes)

<i>Common</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
AutoDisable:	<i>boolean</i>	{read-write}	1.5	Not Supported
CapCompareFirmwareVersion:	<i>boolean</i>	{read-only}	1.9	open
CapPowerReporting:	<i>int32</i>	{read-only}	1.3	open
CapStatisticsReporting:	<i>boolean</i>	{read-only}	1.8	open
CapUpdateFirmware:	<i>boolean</i>	{read-only}	1.9	open
CapUpdateStatistics:	<i>boolean</i>	{read-only}	1.8	open
CheckHealthText:	<i>string</i>	{read-only}	1.5	open
Claimed:	<i>boolean</i>	{read-only}	1.5	open
DataCount:	<i>int32</i>	{read-only}	1.5	Not Supported
DataEventEnabled:	<i>boolean</i>	{read-write}	1.5	Not Supported
DeviceEnabled:	<i>boolean</i>	{read-write}	1.5	open & claim
FreezeEvents:	<i>boolean</i>	{read-write}	1.5	open
OutputID:	<i>int32</i>	{read-only}	1.5	Not Supported
PowerNotify:	<i>int32</i>	{read-write}	1.5	open
PowerState:	<i>int32</i>	{read-only}	1.5	open
State:	<i>int32</i>	{read-only}	1.5	--
DeviceControlDescription:	<i>string</i>	{read-only}	1.5	--
DeviceControlVersion:	<i>int32</i>	{read-only}	1.5	--
DeviceServiceDescription:	<i>string</i>	{read-only}	1.5	open
DeviceServiceVersion:	<i>int32</i>	{read-only}	1.5	open
PhysicalDeviceDescription:	<i>string</i>	{read-only}	1.5	open
PhysicalDeviceName:	<i>string</i>	{read-only}	1.5	open

[Goto Teable 1-17](#)

UPOS Ver1.16 RCSD Specification

Properties (Continued)

<i>Specific</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
CapBatteryCapacityRemaining:	boolean	{read-only}	1.9	open
CapBatteryCapacityRemainingInSeconds:	boolean	{read-only}	1.16	open
CapChargeTime:	boolean	{read-only}	1.16	open
CapFanAlarm:	boolean	{read-only}	1.5	open
CapHeatAlarm:	boolean	{read-only}	1.5	open
CapQuickCharge:	boolean	{read-only}	1.5	open
CapRestartPOS:	boolean	{read-only}	1.9	open
CapShutdownPOS:	boolean	{read-only}	1.5	open
CapStandbyPOS:	boolean	{read-only}	1.9	open
CapSuspendPOS:	boolean	{read-only}	1.9	open
CapTimeMode:	boolean	{read-only}	1.16	open
CapUPSChargeState:	int32	{read-only}	1.5	open
CapVariableBatteryCriticallyLowThreshold:	boolean	{read-only}	1.9	open
CapVariableBatteryCriticallyLowThresholdInSeconds:	boolean	{read-only}	1.16	open
CapVariableBatteryLowThreshold:	boolean	{read-only}	1.9	open
CapVariableBatteryLowThresholdInSeconds:	boolean	{read-only}	1.16	open
BatteryCapacityRemaining:	int32	{read-only}	1.9	open
BatteryCapacityRemainingInSeconds:	int32	{read-only}	1.16	open
BatteryCriticallyLowThreshold:	int32	{read-write}	1.9	open
BatteryCriticallyLowThresholdInSeconds:	int32	{read-write}	1.16	open
BatteryLowThreshold:	int32	{read-write}	1.9	open
BatteryLowThresholdInSeconds:	int32	{read-write}	1.16	open
ChargeTime:	int32	{read-only}	1.16	open
EnforcedShutdownDelayTime:	int32	{read-write}	1.5	open
PowerFailDelayTime:	int32	{read-only}	1.5	open
PowerSource:	int32	{read-only}	1.9	open
QuickChargeMode:	boolean	{read-only}	1.5	open
QuickChargeTime:	int32	{read-only}	1.5	open
TimeMode:	boolean	{read-write}	1.16	open
UPSChargeState:	int32	{read-only}	1.5	open, claim & enable

Goto Table 1-15

UPOS Ver1.16 RCSD Specification

Methods (UML operations)

Common

<i>Name</i>	<i>Version</i>
open (logicalDeviceName: <i>string</i>): void {raises-exception}	1.5
close (): void {raises-exception, use after open}	1.5
claim (timeout: <i>int32</i>): void {raises-exception, use after open}	1.5
release (): void {raises-exception, use after open, claim}	1.5
checkHealth (level: <i>int32</i>): void {raises-exception, use after open, enable}	1.5
clearInput (): void { }	<i>Not supported</i>
clearInputProperties (): void { }	<i>Not supported</i>
clearOutput (): void { }	<i>Not supported</i>
directIO (command: <i>int32</i> , inout data: <i>int32</i> , inout obj: <i>object</i>): void {raises-exception, use after open}	1.5
compareFirmwareVersion (firmwareFileName: <i>string</i> , out result: <i>int32</i>): void {raises-exception, use after open, claim, enable}	1.9
resetStatistics (statisticsBuffer: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.8
retrieveStatistics (inout statisticsBuffer: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.8
updateFirmware (firmwareFileName: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.9
updateStatistics (statisticsBuffer: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.8

Specific

<i>Name</i>	
restartPOS (): void {raises-exception, use after open, enable}	1.9
shutdownPOS (): void {raises-exception, use after open, enable}	1.5
standbyPOS (reason: <i>int32</i>): void {raises-exception, use after open, enable}	1.9
suspendPOS (reason: <i>int32</i>): void {raises-exception, use after open, enable}	1.9

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

<i>Name</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>
upos::events::DataEvent		<i>Not supported</i>	
upos::events::DirectIOEvent			1.5
EventNumber:	<i>int32</i>	{read-only}	
Data:	<i>int32</i>	{read-write}	
Obj:	<i>object</i>	{read-write}	
upos::events::ErrorEvent		<i>Not supported</i>	
upos::events::OutputCompleteEvent		<i>Not supported</i>	
upos::events::StatusUpdateEvent			1.5
Status:	<i>int32</i>	{read-only}	
upos::events::TransitionEvent		<i>Not supported</i>	1.16

[Goto Table 1-17](#)

UPOS Ver1.16 RCSD Specification

General Information

The POS Power programmatic name is “POSPower”.

Capabilities

The POSPower device class has the following capabilities:

- Supports a command to “shut down” the system.
- Supports a command to restart the system.
- Supports a command to “suspend” the system.
- Supports a command to have the system go to standby.
- Supports accessing a power handling mechanism of the underlying operating system and hardware.
- Informs the application if a power fail situation has occurred.
- Informs the application about battery level.
- Informs the application if the UPS charge state has changed.
- Informs the application about high CPU temperature.
- Informs the application about stopped CPU fan.
- Informs the application if an operating system dependent enforced shutdown mechanism is processed.
- Allows the application after saving application data locally or transferring application data to a server to shut down the POS terminal.
- Informs the application about an initiated shutdown.

Device Sharing

The POSPower is a sharable device. Its device sharing rules are:

- After opening and enabling the device, the application may access all properties and methods and will receive status update events.
- If more than one application has opened and enabled the device, all applications may access its properties and methods. Status update events are fired to all of the applications.
- If one application claims the POSPower, then only that application may call the **shutdownPOS**, **standbyPOS**, or **suspendPOS** methods. This feature provides a degree of security, such that these methods may effectively be restricted to the main POS application if that application claims the device at startup.
- See the “Summary” table for precise usage prerequisites.

UPOS Ver1.16 RCSD Specification Model

The general model of POSPower is based on the power model of each device in version 1.3 or later. The same common properties are used but all states relate to the POS terminal itself and not to a peripheral device.

There are three states of the POSPower:

- **ONLINE.** The POS terminal is powered on and ready for use. This is the “operational” state.
- **OFF.** The POS terminal is powered off or detached from the power supplying net. The POS terminal runs on battery power support. This is the powerfail situation.
- **OFFLINE.** The POS terminal is powered on but is running in a “lower-power-consumption” mode. It may need to be placed online by pressing a button or key or something else which may wake up the system.

Power reporting only occurs while the device is open, enabled and power notification is switched on.

In a powerfail situation - that means the POSPower is in the state OFF - the POS terminal will be shut down automatically after the last application has closed the POSPower device or the time specified by the **EnforcedShutdownDelayTime** property has been elapsed.

A call to the **shutdownPOS** method will always shut down the POS terminal independent of the system power state.

Version 1.9 or later

Support of battery powered devices is added. In addition to adding properties to report battery levels and power sources, properties are added to allow for the setting of low and critically low battery levels. The POSPower device also includes the ability to request or respond to request to enter the standby and suspend states. The model does not attempt to duplicate other power management models such as APM and ACPI, but leaves those implementation details to the provider. As a rule, the suspend state will consume less power than the standby state, which in turn will consume less power than the on state. A suggested mapping of these states to other power management models is:

<i>State</i>	<i>ACPI</i>	<i>APM</i>	<i>Description</i>
On	S0	ON	Active, Powered On
Standby	S1	SUSPEND	Displays and drives off, CPU, RAM and fans powered on
Suspend	S3	SUSPEND	Only RAM powered
Off	S5	OFF	Completely powered off

UPOS Ver1.16 RCSD Specification
POSPower Class Diagram

Updated in Release 1.16

The following diagram shows the relationships between the POSPower classes.

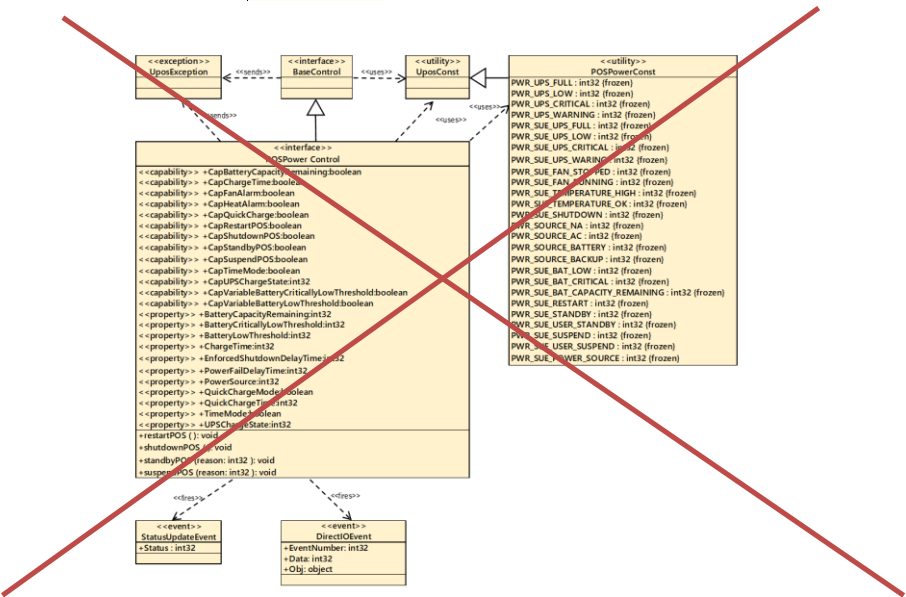
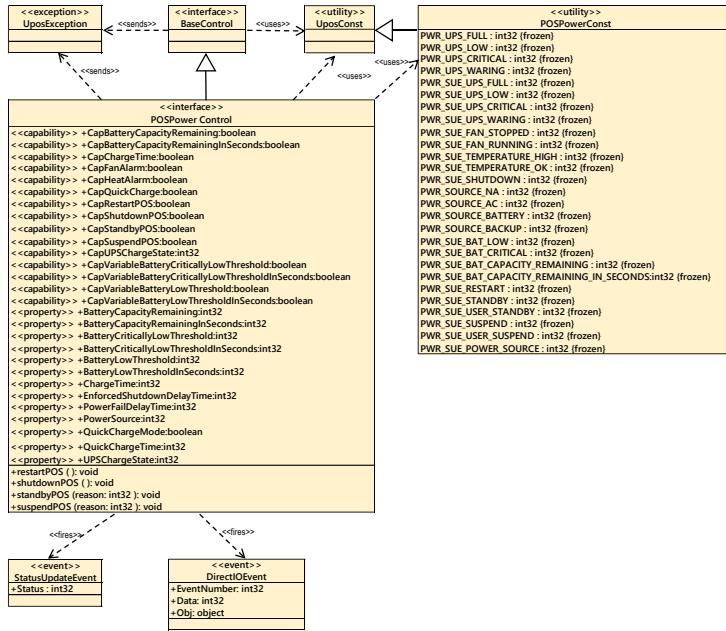


Fig. Chap.29-1 POSPower Class Diagram

[Go to Table 2-2](#)

UPOS Ver1.16 RCSD Specification POSPower Sequence Diagram

The following sequence diagram shows the typical usage of the POSPower device for registering for **StatusUpdateEvents** and an atypical case of initiating a **shutdownPOS** call.

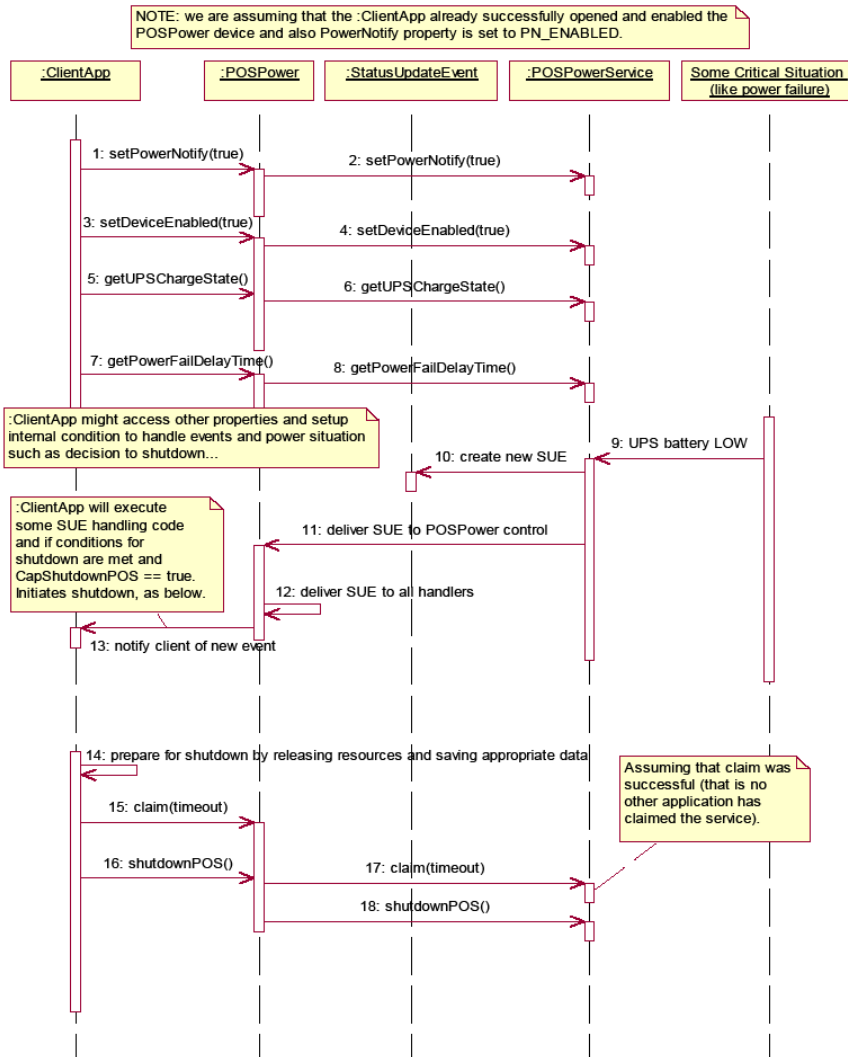


Fig. Chap. 29-2 POSPower Sequence Diagram

UPOS Ver1.16 RCSD Specification

POSPower Standby Sequence Diagram

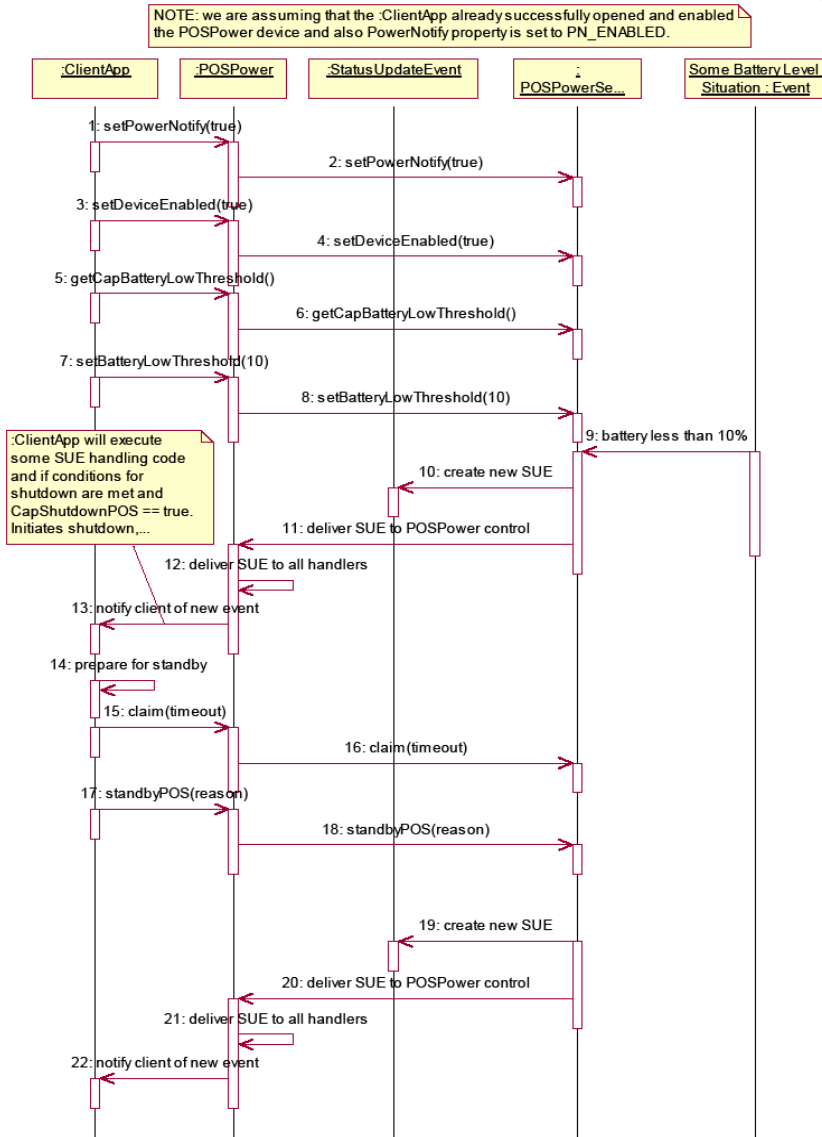


Fig. Chap. 29-3 POSPower Standby Sequence Diagram

UPOS Ver1.16 RCSD Specification POSPower State Diagram

The following state diagram depicts the POSPower Control device model.

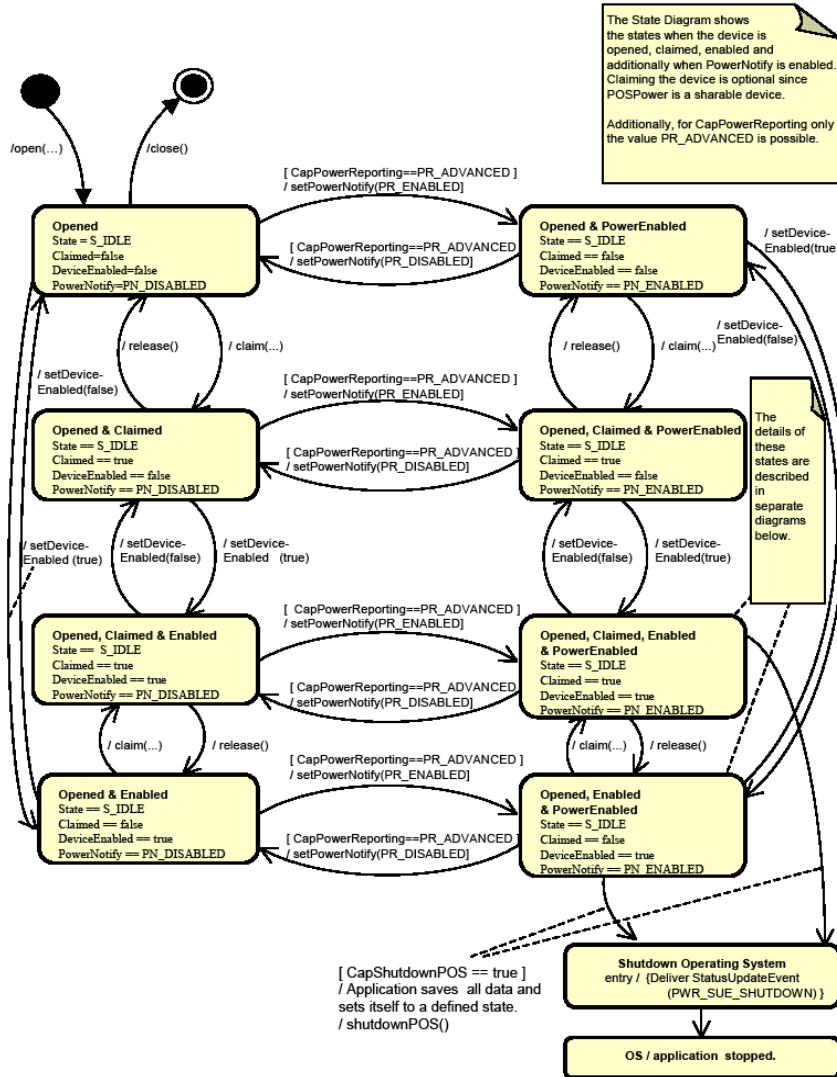


Fig. Chap. 29-4 Power State Diagram (POSPower Control Device Model)

[Goto Table 2-3](#)

POSPower PowerState Diagram - Part 1

The following state diagram depicts the POSPower Power States.

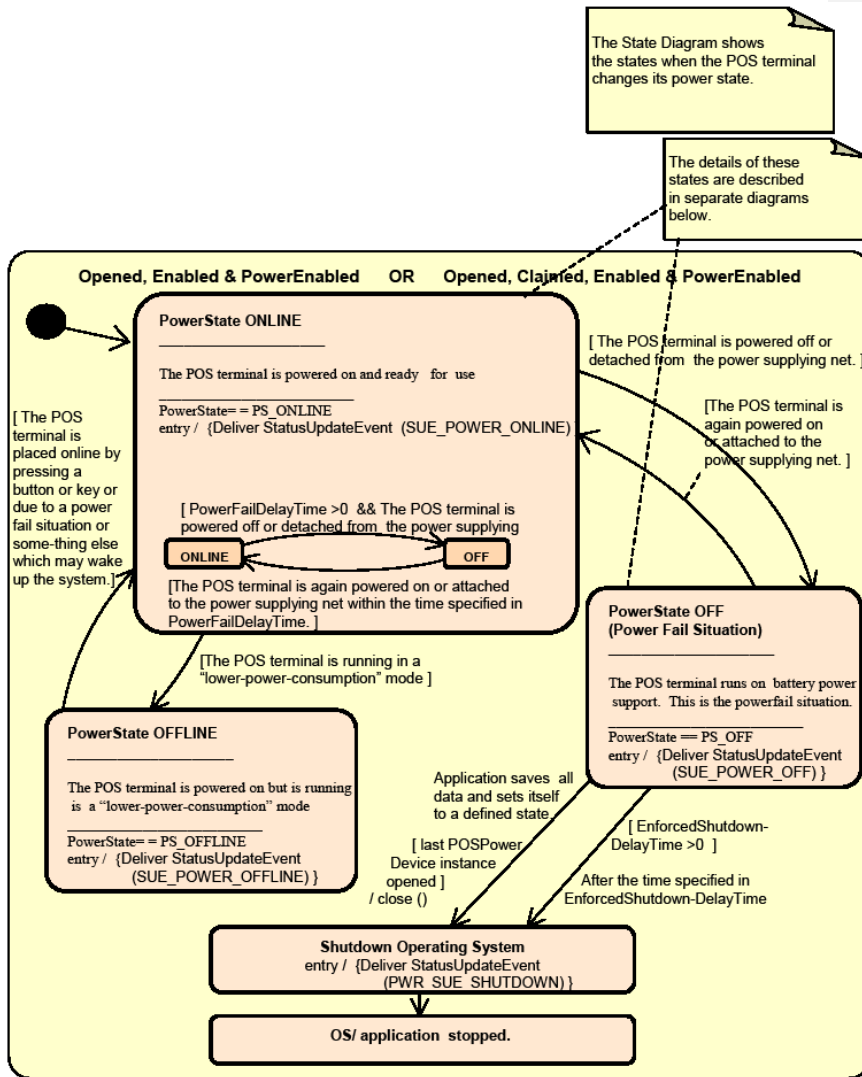


Fig. Chap. 29-5 POSPower PowerState Diagram (Part 1)

UPOS Ver1.16 RCSD Specification
POSPower PowerState Diagram - Part 2

The following state diagram depicts the POSPower PowerState ONLINE.

The State Diagram shows the sub states in the PowerState ONLINE state when charging the UPS battery.

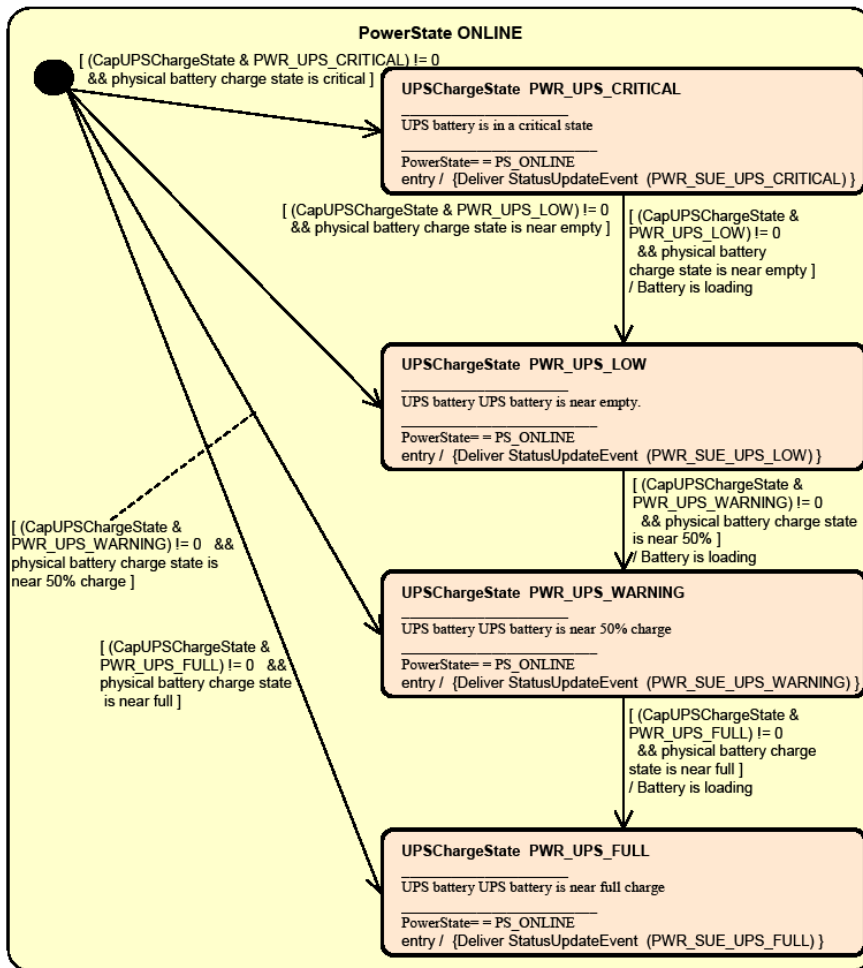


Fig. Chap. 29-6 POSPower PowerState Diagram (Part 2)

UPOS Ver1.16 RCSD Specification
POSPower PowerState Diagram - Part 3

The following state diagram depicts the POSPower PowerState OFF.

The State Diagram shows the sub states in the PowerState OFF state when unloading the UPS battery.

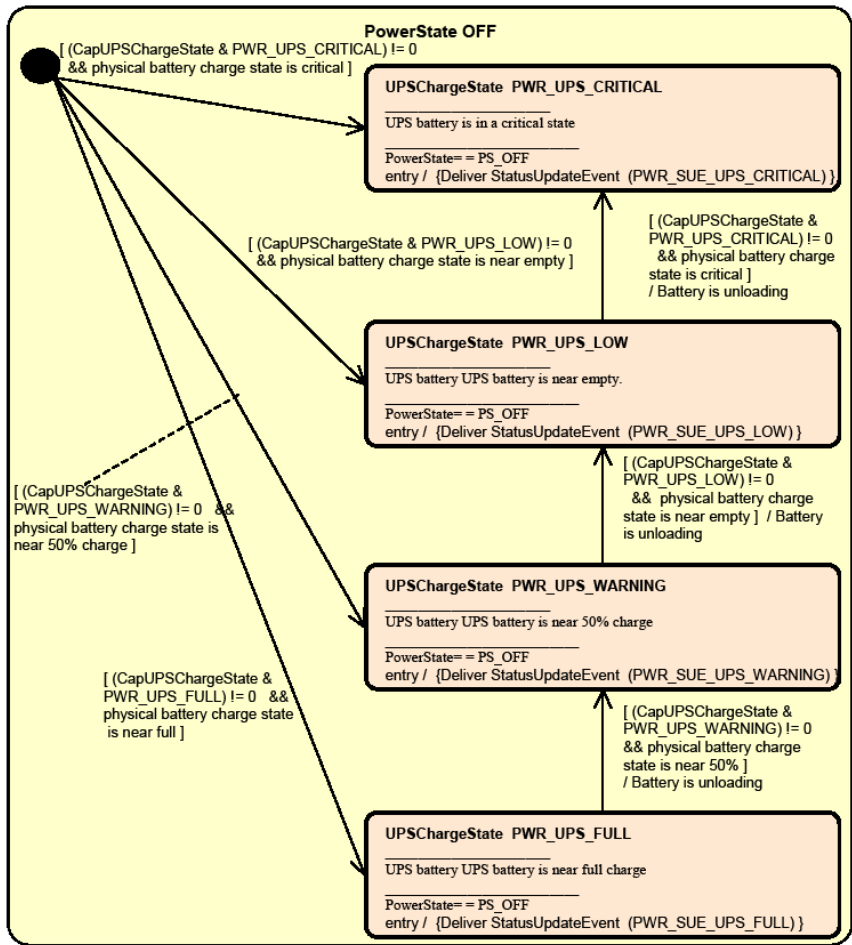


Fig. Chap. 29-7 POSPower PowerState Diagram (Part 3)

UPOS Ver1.16 RCSD Specification
POSPower State Chart Diagram for Fan and Temperature

The following state diagram depicts the handling of fan and temperature alarms.

The State Diagrams shows the states for handling high CPU temperature and stopped CPU fan.

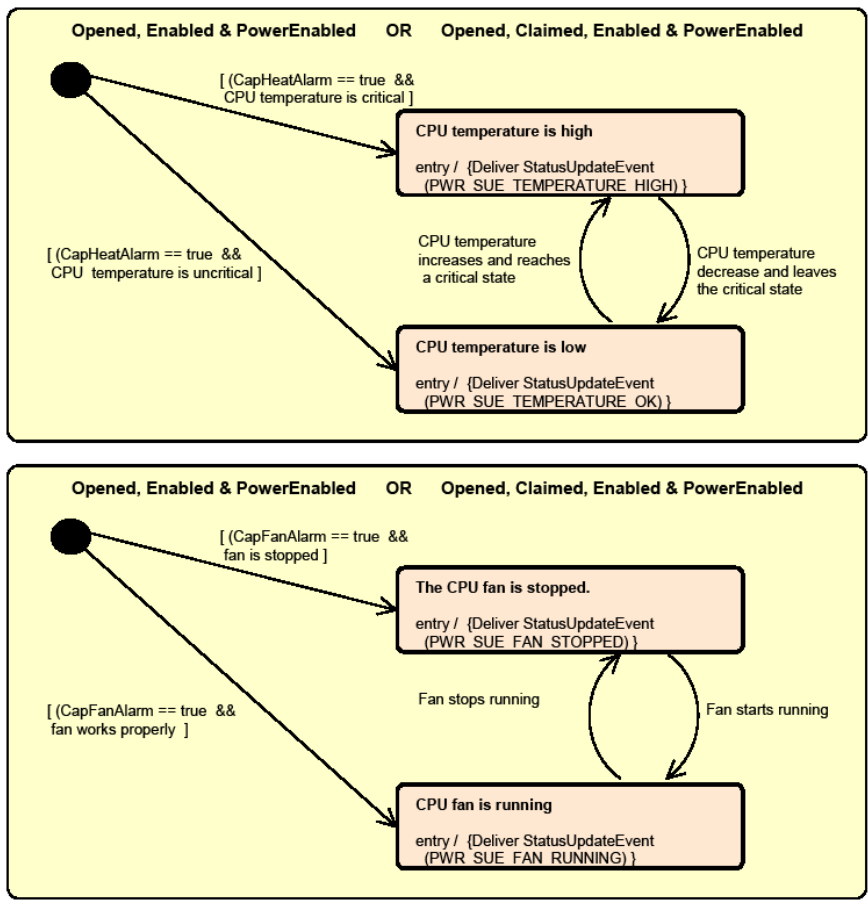


Fig. Chap. 29-8 POSPower State Chart Diagram (Fan and Temperature)

UPOS Ver1.16 RCSD Specification

POSPower Battery State Diagram

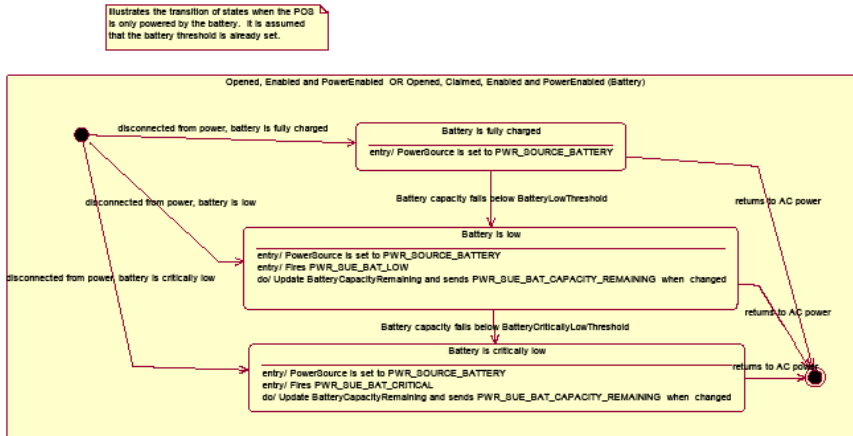


Fig. Chap. 29-9 POSPower Battery State Diagram

UPOS Ver1.16 RCSD Specification
POSPower Power Transitions State Diagram

The state diagram illustrates the changes when the POS is powered by battery

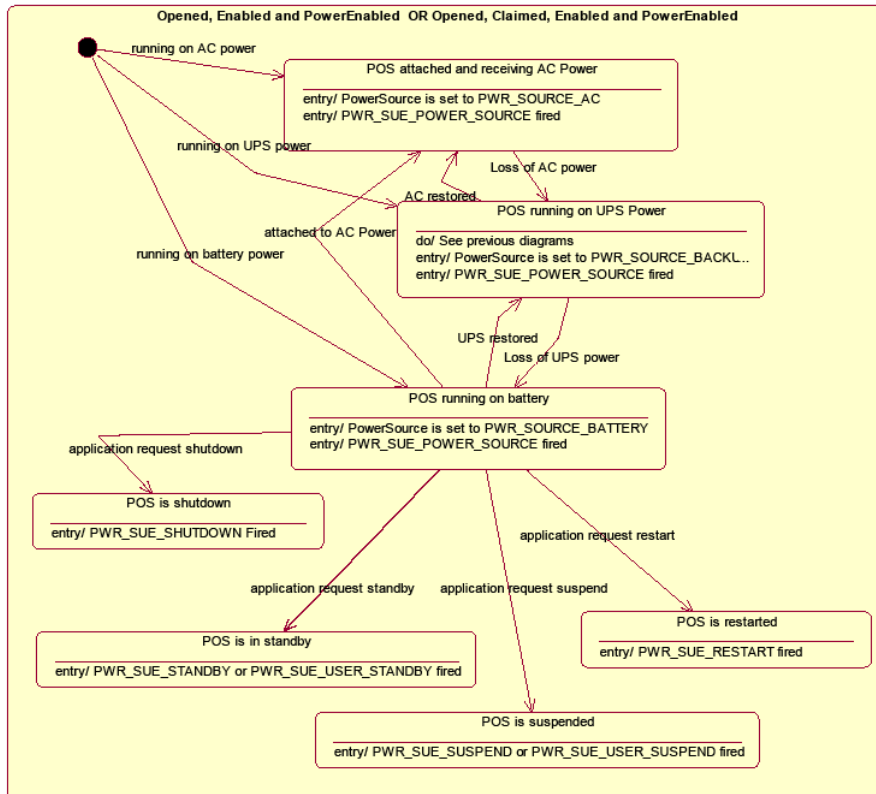


Fig. Chap. 29-10 POSPower Power Transitions State Diagram

UPOS Ver1.16 RCSD Specification

Properties (UML attributes)

BatteryCapacityRemaining Property

Syntax	BatteryCapacityRemaining: <i>int32</i> {read-only, access after open}
Remarks	A value of 0 to 100 represents percent of battery capacity remaining. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	CapBatteryCapacityRemaining Property

BatteryCapacityRemainingInSeconds Property *Added in Release 1.16*

Syntax	BatteryCapacityRemainingInSeconds: <i>int32</i> {read-only, access after open}
Remarks	A value of battery capacity remaining in seconds. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	CapBatteryCapacityRemainingInSeconds Property

[Goto Table1-18](#)

BatteryCriticallyLowThreshold Property

Syntax	BatteryCriticallyLowThreshold: <i>int32</i> {read-write, access after open}
Remarks	If not zero, this property holds the threshold at which a PWR_SUE_BAT_CRITICAL StatusUpdateEvent is generated. The values 1 through 99 represent the percentage of the capacity remaining. The value 0 indicates that Battery Critically Low reporting is not supported or is disabled. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	CapVariableBatteryCriticallyLowThreshold Property, StatusUpdateEvent

UPOS Ver1.16 RCSD Specification

BatteryCriticallyLowThresholdInSeconds Property

Added in Release 1.16

Syntax	BatteryCriticallyLowThresholdInSeconds: <i>int32</i> {read-write, access after open}
Remarks	If not zero, this property holds the threshold at which a PWR_SUE_BAT_CRITICAL StatusUpdateEvent is generated. The values of seconds of the capacity remaining. The value 0 indicates that Battery Critically Low reporting is not supported or is disabled. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	CapVariableBatteryCriticallyLowThresholdInSeconds Property, StatusUpdateEvent

[Goto Table 1-19](#)

BatteryLowThreshold Property

Syntax	BatteryLowThreshold: <i>int32</i> {read-write, access after open}
Remarks	If not zero, this property holds the threshold at which a PWR_SUE_BAT_LOW StatusUpdateEvent is generated. The value 1 to 99 represents the percent capacity remaining. The value 0 indicates that battery low reporting is not supported or is disabled. If variable battery low threshold is supported, setting a value between 1 and 99 sets the threshold to that value. Setting a value of zero disables battery low reporting. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	CapVariableBatteryLowThreshold Property, StatusUpdateEvent

BatteryLowThresholdInSeconds Property

Added in Release 1.16

Syntax	BatteryLowThresholdInSeconds: <i>int32</i> {read-write, access after open}
Remarks	If not zero, this property holds the threshold at which a PWR_SUE_BAT_LOW StatusUpdateEvent is generated. The value of seconds of the capacity remaining. The value 0 indicates that battery low reporting is not supported or is disabled. If variable battery low threshold is supported, setting a value of seconds sets the threshold to that value. Setting a value of zero disables battery low reporting. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	CapVariableBatteryLowThresholdInSeconds Property, StatusUpdateEvent

[Goto Table 1-20](#)

UPOS Ver1.16 RCSD Specification

CapBatteryCapacityRemaining Property

Syntax	CapBatteryCapacityRemaining: <i>boolean</i> {read-only, access after open}
Remarks	If true, the device is able to provide battery capacity information. Otherwise it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	BatteryCapacityRemaining Property

CapBatteryCapacityRemainingInSeconds Property

Added in Release 1.16

Syntax	CapBatteryCapacityRemainingInSeconds : <i>boolean</i> {read-only, access after open}
Remarks	If true, the device is able to provide battery capacity information seconds. Otherwise it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	BatteryCapacityRemainingInSeconds Property

[Goto Table 1-21](#)

CapChargeTime Property

Added in Release 1.16

Syntax	CapChargeTime: <i>boolean</i> {read-only, access after open}
Remarks	If true, the device is able to acquire the remaining time until full charging. Otherwise it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	ChargeTime Property.

CapFanAlarm Property

Syntax	CapFanAlarm: <i>boolean</i> {read-only, access after open}
Remarks	If true, the device is able to detect whether the CPU fan is stopped. Otherwise it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.

UPOS Ver1.16 RCSD Specification

CapHeatAlarm Property

Syntax	CapHeatAlarm: <i>boolean</i> {read-only, access after open}
Remarks	If true the device is able to detect whether the CPU is running at too high of a temperature. Otherwise it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.

CapQuickCharge Property

Syntax	CapQuickCharge: <i>boolean</i> {read-only, access after open}
Remarks	If true, the power management allows the charging of the UPS battery in quick mode. The time for charging the battery is shorter than usual. Otherwise it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	QuickChargeMode Property, QuickChargeTime Property.

CapRestartPOS Property

Syntax	CapRestartPOS: <i>boolean</i> {read-only, access after open}
Remarks	If true the device is able to explicitly restart the POS. Otherwise it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	restartPOS Method.

CapShutdownPOS Property

Syntax	CapShutdownPOS: <i>boolean</i> {read-only, access after open}
Remarks	If true the device is able to explicitly shut down the POS. Otherwise it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	shutdownPOS Method.

UPOS Ver1.16 RCSD Specification CapStandbyPOS Property

- Syntax** CapStandbyPOS: *boolean* {read-only, access after open}
- Remarks** If true, the device is able to request that the POS System enter the Standby state. Otherwise it is false.
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
- See Also** standbyPOS Method.

CapSuspendPOS Property

- Syntax** CapSuspendPOS: *boolean* {read-only, access after open}
- Remarks** If true, the device is able to request that the POS System enter the Suspend state. Otherwise it is false.
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
- See Also** suspendPOS Method.

~~CapTimeMode Property~~ Added in Release 1.16

- ~~**Syntax** CapTimeMode: *boolean* {read-only, access after open}~~
- ~~**Remarks** If true the device is able to switch the unit of battery remaining / threshold related property value to seconds. Otherwise it is false.
This property is initialized by the **open** method.~~
- ~~**Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.~~
- ~~**See Also** [TimeMode Property](#)~~ [Goto Table 1-22](#)

Formatted: Space After: 4 pt, Don't keep with next

CapUPSChargeState Property

- Syntax** CapUPSChargeState: *int32* {read-only, access after open}
- Remarks** If not equal to zero, the UPS can deliver one or more charge states. It can contain any of the following values logically ORed together.
- | Value | Meaning |
|------------------|--|
| PWR_UPS_FULL | UPS battery is near full charge. |
| PWR_UPS_WARNING | UPS battery is near 50% charge. |
| PWR_UPS_LOW | UPS battery is near empty. Application shutdown should be started to ensure that is can be completed before the battery charge is depleted. A minimum of 2 minutes of normal system operation can be assumed when this state is entered unless this is the first state reported upon entering the “Off” power state. |
| PWR_UPS_CRITICAL | UPS battery is in a critical state and could be disconnected at any time without further warning. This property is initialized by the open method. |
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
- See Also** UPSChargeState Property.

UPOS Ver1.16 RCSD Specification

CapVariableBatteryCriticallyLowThreshold Property

Syntax	CapVariableBatteryCriticallyLowThreshold: <i>boolean</i> {read-only, access after open}
Remarks	If true, the device supports a variable threshold for critically low battery. Otherwise it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	BatteryCriticallyLowThreshold Property, StatusUpdateEvent

CapVariableBatteryCriticallyLowThresholdInSeconds Property

Added in Release 1.16

Syntax	CapVariableBatteryCriticallyLowThresholdInSeconds: <i>boolean</i> {read-only, access after open}
Remarks	If true, the device supports a second’s variable threshold for critically low battery. Otherwise it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	BatteryCriticallyLowThresholdInSeconds Property, StatusUpdateEvent

[Goto Table 1-23](#)

CapVariableBatteryLowThreshold Property

Syntax	CapVariableBatteryLowThreshold: <i>boolean</i> {read-only, access after open}
Remarks	If true, the device supports a variable threshold for battery low. Otherwise it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	BatteryLowThreshold Property, StatusUpdateEvent

CapVariableBatteryLowThresholdInSeconds Property

Added in Release 1.16

Syntax	CapVariableBatteryLowThresholdInSeconds: <i>boolean</i> {read-only, access after open}
Remarks	If true, the device supports a second’s variable threshold for battery low. Otherwise it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	BatteryLowThresholdInSeconds Property, StatusUpdateEvent

[Goto Table 1-24](#)

UPOS Ver1.16 RCSD Specification

ChargeTime Property

Added in Release 1.16

- Syntax** **ChargeTime: int32** {read-only, access after open}
- Remarks** Indicates the time remaining until the battery is fully charged in seconds.
If equal to zero the battery is not charging or not supported.
This property is only set if **CapChargeTime** is true.
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
- See Also** **CapChargeTime** Property.

EnforcedShutdownDelayTime Property

- Syntax** **EnforcedShutdownDelayTime: int32** {read-write, access after open}
- Remarks** If not equal to zero the system has a built-in mechanism to shut down the POS terminal after a determined time in a power fail situation. This property contains the time in milliseconds when the system will shut down automatically after a power failure. A power failure is the situation when the POS terminal is powered off or detached from the power supplying net and runs on UPS.
If zero no automatic shutdown is performed and the application has to call itself the **shutdownPOS** method.
Applications will be informed about an initiated automatic shutdown. This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
- See Also** **shutdownPOS** Method.

PowerFailDelayTime Property

- Syntax** **PowerFailDelayTime: int32** {read-only, access after open}
- Remarks** This property contains the time in milliseconds for power fail intervals which will not create a power fail situation. In some countries the power has sometimes short intervals where the power supply is interrupted. Those short intervals are in the range of milliseconds up to a few seconds and are handled by batteries or other electric equipment and should not cause a power fail situation. The power fail interval starts when the POS terminal is powered off or detached from the power supplying net and runs on UPS. The power fail interval ends when the POS terminal is again powered on or attached to the power supplying net. However, if the power fail interval is longer than the time specified in the **PowerFailDelayTime** property a power fail situation is created.

Usually this parameter is a configuration parameter of the underlying power management. So, the application can only read this property.

This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.

UPOS Ver1.16 RCSD Specification

PowerSource Property

Syntax	PowerSource: <i>int32</i> {read-only, access after open}										
Remarks	This property holds the current power source if power source reporting is available. A StatusUpdateEvent is generated each time this property is updated. <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>PWR_SOURCE_NA</td><td>Power source reporting is not available.</td></tr><tr><td>PWR_SOURCE_AC</td><td>The current power source is the AC line.</td></tr><tr><td>PWR_SOURCE_BATTERY</td><td>The current power source is a system battery. This value is only presented for systems that operate normally on battery.</td></tr><tr><td>PWR_SOURCE_BACKUP</td><td>The current power source is a backup source such as an UPS or backup battery.</td></tr></tbody></table> <p>This property is initialized by the open method.</p>	<u>Value</u>	<u>Meaning</u>	PWR_SOURCE_NA	Power source reporting is not available.	PWR_SOURCE_AC	The current power source is the AC line.	PWR_SOURCE_BATTERY	The current power source is a system battery. This value is only presented for systems that operate normally on battery.	PWR_SOURCE_BACKUP	The current power source is a backup source such as an UPS or backup battery.
<u>Value</u>	<u>Meaning</u>										
PWR_SOURCE_NA	Power source reporting is not available.										
PWR_SOURCE_AC	The current power source is the AC line.										
PWR_SOURCE_BATTERY	The current power source is a system battery. This value is only presented for systems that operate normally on battery.										
PWR_SOURCE_BACKUP	The current power source is a backup source such as an UPS or backup battery.										
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.										
See Also	StatusUpdateEvent										

QuickChargeMode Property

Syntax	QuickChargeMode: <i>boolean</i> {read-only, access after open}
Remarks	If true, the UPS battery is being recharged in a quick charge mode. If false, it is being charged in a normal mode. <p>This property is only set if CapQuickCharge is true.</p>
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	CapQuickCharge Property, QuickChargeTime Property.

QuickChargeTime Property

Syntax	QuickChargeTime: <i>int32</i> {read-only, access after open}
Remarks	This time specifies the remaining time for charging the UPS battery in quick charge mode. After the time has elapsed, the UPS battery charging mechanism of power management usually switches into normal mode. <p>This time is specified in milliseconds.</p> <p>This property is only set if CapQuickCharge is true.</p>
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	CapQuickCharge Property, QuickChargeTime Property.

UPOS Ver1.16 RCSD Specification

~~TimeMode Property~~

~~-Added in Release 1.16~~

~~**Syntax** UPSChargeState: *boolean* {read-write, access after open}~~
~~**Remarks** If true, the value of the battery remaining / threshold related property is in seconds. If false, the value of the battery remaining / threshold related property is in percent. This property is initialized by the open method.~~
~~**Errors** A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20~~
~~**See Also** CapTimeMode Property, BatteryCapacityRemaining Property, BatteryCriticallyLowThreshold Property, BatteryLowThreshold Property.~~

[Goto Table 1-22](#)

Formatted: Indent: Hanging: 0.76", Line spacing: At least 12 pt, Tab stops: 1.25", Left

UPSChargeState Property

Syntax UPSChargeState: *int32* {read-only, access after open, enable}

Remarks This property holds the actual UPS charge state.

It has one of the following values:

Value	Meaning
PWR_UPS_FULL	UPS battery is near full charge.
PWR_UPS_WARNING	UPS battery is near 50% charge.
PWR_UPS_LOW	UPS battery is near empty. Application shutdown should be started to ensure that it can be completed before the battery charge is depleted. A minimum of 2 minutes of normal system operation can be assumed when this state is entered unless this is the first state reported upon entering the "Off" power state.
PWR_UPS_CRITICAL	UPS battery is in a critical state and could be disconnected at any time without further warning.

This property is initialized and kept current while the device is enabled.

Errors A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20

See Also CapUPSChargeState Property.

UPOS Ver1.16 RCSD Specification

Methods (UML operations)

restartPOS Method

Syntax restartPOS ():void {raises-exception, use after open, enable}

Remarks Call to restart the POS terminal. This method will always restart the system independent of the system power state.

If the POSPower is claimed, only the application which claimed the device is able to restart the POS terminal.

Applications will be informed about an initiated restart.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20

Some possible values of the exception’s *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	This method is not supported (see the CapRestartPOS property)

See Also **CapRestartPOS** Property

shutdownPOS Method

Syntax shutdownPOS ():void {raises-exception, use after open, enable}

Remarks Call to shut down the POS terminal. This method will always shut down the system independent of the system power state.

If the POSPower is claimed, only the application which claimed the device is able to shut down the POS terminal.

Applications will be informed about an initiated shutdown.

It is recommended that in a power fail situation an application has to call this method after saving all data and setting the application to a defined state.

If the **EnforcedShutdownDelayTime** property specifies a time greater than zero and the application did not call the **shutdownPOS** method within the time specified in **EnforcedShutdownDelayTime**, the system will be shut down automatically. This mechanism may be provided by an underlying operating system to prevent the battery from being emptied before the system is shut down.

This method is only supported if **CapShutdownPOS** is true.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20

Some possible values of the exception’s *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	This method is not supported. (See the CapShutdownPOS property)

See Also **CapShutdownPOS** Property, **EnforcedShutdownDelayTime** Property.

UPOS Ver1.16 RCSD Specification

standbyPOS Method

Syntax **standbyPOS (reason: int32):**
 void {raises-exception, use after open, enable}

Remarks Call to request that the system be placed into the Standby state or to respond to a request from the system, OS or other application that the system be put into Standby state.

The *reason* parameter indicates the reason the POS terminal should enter a standby state:

<u>Value</u>	<u>Description</u>
PWR_REASON_REQUEST	Call is to request that the system enter the standby state.
PWR_REASON_ALLOW	Call is a response to a standby Status Update Event and specifies that the request should be allowed.
PWR_REASON_DENY	Call is a response to a standby Status Update Event and specifies that the request should be denied.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20

Some possible values of the exception’s *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	This method is not supported (see the CapStandbyPOS property)

See Also **CapStandbyPOS** Property.

suspendPOS Method

Syntax **suspendPOS (reason: int32):**
 void {raises-exception, use after open, enable}

Remarks Call to request that the system be placed into the Suspend state or to respond to a request from the system, OS or other application that the system be put into Suspend state.

The *reason* parameter indicates the reason the POS terminal should enter a standby state:

<u>Value</u>	<u>Description</u>
PWR_REASON_REQUEST	Call is to request that the system enter the suspend state.
PWR_REASON_ALLOW	Call is a response to a suspend Status Update Event and specifies that the request should be allowed.
PWR_REASON_DENY	Call is a response to a suspend Status Update Event and specifies that the request should be denied.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20

Some possible values of the exception’s *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	This method is not supported (see the CapSuspendPOS property)

See Also **CapSuspendPOS** Property.

UPOS Ver1.16 RCSD Specification
Events (UML Interfaces)

DirectIOEvent

```
<<event>> upos::events::DirectIOEvent
EventNumber : int32 {read-only}
Data : int32 {read-write}
Obj : object{read-write}
```

Description Provides Service information directly to the application. This event provides a means for a vendor specific POSPower Service to provide events to the application that are not otherwise supported by the ~~Control~~ device control.

Attributes This event contains the following attributes:

Attributes	Type	Description
<i>EventNumber</i>	<i>int32</i>	Event number whose specific values are assigned by the Service.
<i>Data</i>	<i>int32</i>	Additional numeric data. Specific values vary by the <i>EventNumber</i> and the Service. This property is settable.
<i>Obj</i>	<i>object</i>	Additional data whose usage varies by the <i>EventNumber</i> and Service. This property is settable.

Remarks This event is to be used only for those types of vendor specific functions that are not otherwise described. Use of this event may restrict the application program from being used with other vendor's POSPower devices which may not have any knowledge of the Service's need for this event.

See Also "Errors" on page Intro-20, **directIO** Method. [Goto Table 1-25](#)

StatusUpdateEvent

```
<<event>> upos::events::StatusUpdateEvent
Status : int32 {read-only}
```

Description Delivered when **UPSChargeState** changes or an alarm situation occurs.

Attributes This event contains the following attribute:

Attributes	Type	Description
<i>Status</i>	<i>int32</i>	See below.

The *Status* property contains the updated power status or alarm status.

Value	Meaning
PWR_SUE_UPS_FULL	UPS battery is near full charge. Can be returned if CapUPSChargeState contains PWR_UPS_FULL.
PWR_SUE_UPS_WARNING	UPS battery is near 50% charge. Can be returned if CapUPSChargeState contains PWR_UPS_WARNING.
PWR_SUE_UPS_LOW	UPS battery is near empty. Application shutdown should be started to ensure that it can be completed before the battery charge is depleted. A minimum of 2 minutes of normal system operation can be assumed when this state is entered unless this is the first charge state reported upon entering the "Off" state. Can be returned if CapUPSChargeState contains PWR_UPS_LOW.
PWR_SUE_UPS_CRITICAL	

UPOS Ver1.16 RCSD Specification

	UPS is in critical state, and will in short time be disconnected. Can be returned if CapUPSChargeState contains PWR_UPS_CRITICAL.
PWR_SUE_FAN_STOPPED	The CPU fan is stopped. Can be returned if CapFanAlarm is true.
PWR_SUE_FAN_RUNNING	The CPU fan is running. Can be returned if CapFanAlarm is true.
PWR_SUE_TEMPERATURE_HIGH	The CPU is running on high temperature. Can be returned if CapHeatAlarm is true.
PWR_SUE_TEMPERATURE_OK	The CPU is running on normal temperature. Can be returned if CapHeatAlarm is true.
PWR_SUE_SHUTDOWN	The system will shut down immediately.
PWR_SUE_BAT_LOW	The system remaining battery capacity is at or below the low battery threshold and the system is operating from the battery.
PWR_SUE_BAT_CRITICAL	The system remaining battery capacity is at or below the critically low battery threshold and the system is operating from the battery.
PWR_SUE_BAT_CAPACITY_REMAINING.	The BatteryCapacityRemaining property has been updated
PWR_SUE_BAT_CAPACITY_REMAINING_IN_SECONDS	The BatteryCapacityRemainingInSeconds property has been updated
PWR_SUE_RESTART	The system will restart immediately.
PWR_SUE_STANDBY	The system is requesting a transition to the Standby state
PWR_SUE_USER_STANDBY	The system is requesting a transition to the Standby state as a result of user input.
PWR_SUE_SUSPEND	The system is requesting a transition to the Suspend state.
PWR_SUE_USER_SUSPEND	The system is requesting a transition to the Suspend state as a result of user input.
PWR_SUE_PWR_SOURCE	The PowerSource property has been updated.

*Note that Release 1.3 added Power State Reporting with additional Power reporting **StatusUpdateEvent** values.*

UPOS Ver1.16 RCSD Specification

The Update Firmware capability, added in *Release 1.9*, added additional *Status* values for communicating the status/progress of an asynchronous update firmware process. See “**StatusUpdateEvent**” description on page 1-34.

See Also **CapFanAlarm** Property, **CapHeatAlarm** Property, **CapUPSChargeState** Property, **UPSChargeState** Property.

Video Capture

This Chapter defines the Video Capture device category.

Summary

Properties (UML attributes)

<i>Common</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
AutoDisable:	<i>boolean</i>	{read-write}	1.16	Not supported open
CapCompareFirmwareVersion:	<i>boolean</i>	{read-only}	1.16	open
CapPowerReporting:	<i>int32</i>	{read-only}	1.16	open
CapStatisticsReporting:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateFirmware:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateStatistics:	<i>boolean</i>	{read-only}	1.16	open
CheckHealthText:	<i>string</i>	{read-only}	1.16	open
Claimed:	<i>boolean</i>	{read-only}	1.16	open
DataCount:	<i>int32</i>	{read-only}	1.16	Not supported open
DataEventEnabled:	<i>boolean</i>	{read-write}	1.16	Not supported open
DeviceEnabled:	<i>boolean</i>	{read-write}	1.16	open & claim
FreezeEvents:	<i>boolean</i>	{read-write}	1.16	open
OutputID:	<i>int32</i>	{read-only}	1.16	Not supported
PowerNotify:	<i>int32</i>	{read-write}	1.16	open
PowerState:	<i>int32</i>	{read-only}	1.16	open
State:	<i>int32</i>	{read-only}	1.16	--
DeviceControlDescription:	<i>string</i>	{read-only}	1.16	--
DeviceControlVersion:	<i>int32</i>	{read-only}	1.16	--
DeviceServiceDescription:	<i>string</i>	{read-only}	1.16	open
DeviceServiceVersion:	<i>int32</i>	{read-only}	1.16	open
PhysicalDeviceDescription:	<i>string</i>	{read-only}	1.16	open
PhysicalDeviceName:	<i>string</i>	{read-only}	1.16	open

[Goto Table1-26](#)

UPOS Ver1.16 RCSD Specification

Properties (Continued)

<i>Specific</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
CapAssociatedHardTotalsDevice:	string	{read-only}	1.16	open
CapCameraAutoExposure:	boolean	{read-only}	1.16	open
CapCameraAutoFocus:	boolean	{read-only}	1.16	open
CapCameraAutoGain:	boolean	{read-only}	1.16	open
CapCameraAutoWhiteBalance:	boolean	{read-only}	1.16	open
CapCameraBrightness:	boolean	{read-only}	1.16	open
CapCameraContrast:	boolean	{read-only}	1.16	open
CapCameraExposure:	boolean	{read-only}	1.16	open
CapCameraGain:	boolean	{read-only}	1.16	open
CapCameraHorizontalFlip:	boolean	{read-only}	1.16	open
CapCameraHue:	boolean	{read-only}	1.16	open
CapCapture:	boolean	{read-only}	1.16	open
CapCaptureColorSpace:	boolean	{read-only}	1.16	open
CapCaptureColorSpaceList:	string	{read-only}	1.16	open
CapCaptureFrameRate:	boolean	{read-only}	1.16	open
CapCaptureMaxFrameRate:	int32	{read-only}	1.16	open
CapCaptureResolution:	boolean	{read-only}	1.16	open
CapCaptureResolutionList:	string	{read-only}	1.16	open
CapDecodeData:	boolean	{read-only}	1.16	open
CapIndividualRecognition:	boolean	{read-only}	1.16	open
CapPhotograph:	boolean	{read-only}	1.16	open
CapPhotoColorSpace:	boolean	{read-only}	1.16	open
CapPhotoFrameRate:	boolean	{read-only}	1.16	open
CapPhotographResolution:	boolean	{read-only}	1.16	open
CapPhotographResolutionList:	boolean	{read-only}	1.16	open
CapPhotographType:	boolean	{read-only}	1.16	open
CapPhotographTypeList:	boolean	{read-only}	1.16	open
CapCameraSaturation:	boolean	{read-only}	1.16	open
CapStorage:	int32	{read-only}	1.16	open
CapCameraVerticalFlip:	boolean	{read-only}	1.16	open
CapVideoRecording:	boolean	{read-only}	1.16	open
CapVideoColorSpace:	boolean	{read-only}	1.16	open

[Goto Table 1-27](#)

[Goto Table 1-28](#)

UPOS Ver1.16 RCSD Specification

CapVideoRecordingFrameRate:	<i>boolean</i>	{read-only}	1.16	open
CapVideoRecordingMaxFrameRate:	<i>int32</i>	{read-only}	1.16	open
CapVideoRecordingResolution:	<i>boolean</i>	{read-only}	1.16	open
CapVideoRecordingType:	<i>boolean</i>	{read-only}	1.16	open
CapVideoRecordingResolutionList:	<i>string</i>	{read-only}	1.16	open
BarCodeEnabled:	<i>boolean</i>	{read-write}	1.16	open, claim & enable
CameraAutoExposition:	<i>boolean</i>	{read-write}	1.16	open, claim & enable
CameraAutoFocus:	<i>boolean</i>	{read-write}	1.16	open, claim & enable
CameraAutoGain:	<i>boolean</i>	{read-write}	1.16	open, claim & enable
CameraAutoWhiteBalance:	<i>boolean</i>	{read-write}	1.16	open, claim & enable
CameraBrightness:	<i>int32</i>	{read-write}	1.16	open, claim & enable
CameraContrast:	<i>int32</i>	{read-write}	1.16	open, claim & enable
CameraExposure:	<i>int32</i>	{read-write}	1.16	open, claim & enable
CameraGain:	<i>int32</i>	{read-write}	1.16	open, claim & enable
CameraHorizontalFlip:	<i>boolean</i>	{read-write}	1.16	open, claim & enable
CameraHue:	<i>int32</i>	{read-write}	1.16	open, claim & enable
CapturePhotoColorSpace:	<i>string</i>	{read-write}	1.16	open, claim & enable
PhotoColorSpaceList:	<i>string</i>	{read-only}	1.16	open
CapturePhotoFrameRate:	<i>int32</i>	{read-write}	1.16	open, claim & enable
PhotoMaxFrameRate:	<i>int32</i>	{read-only}	1.16	open
CapturePhotoResolution:	<i>string</i>	{read-write}	1.16	open, claim & enable
PhotographResolution:	<i>string</i>	{read-write}	1.16	open, claim & enable
PhotoResolutionList:	<i>string</i>	{read-only}	1.16	open
IndividualRecognitionEnabled:	<i>boolean</i>	{read-write}	1.16	open, claim & enable
PhotographType:	<i>string</i>	{read-write}	1.16	open, claim & enable
PhotoTypeList:	<i>string</i>	{read-only}	1.16	open
RemainingRecordingTimeInSec:	<i>int32</i>	{read-only}	1.16	open, claim & enable
CameraSaturation:	<i>int32</i>	{read-write}	1.16	open, claim & enable
Storage:	<i>int32</i>	{read-write}	1.16	open, claim & enable
CameraVerticalFlip:	<i>boolean</i>	{read-write}	1.16	open, claim & enable
VideoCaptureMode:	<i>int32</i>	{read-only}	1.16	open, claim & enable
VideoColorSpace:	<i>string</i>	{read-write}	1.16	open, claim & enable
VideoColorSpaceList:	<i>string</i>	{read-only}	1.16	open

[Goto Table 1-29](#)
[Goto Table 1-31](#)

UPOS Ver1.16 RCSD Specification

VideoRecordingFrameRate:	<i>int32</i>	{read-write}	1.16	open, claim & enable
VideoMaxFrameRate:	<i>int32</i>	{read-only}	1.16	open
VideoRecordingResolution:	<i>string</i>	{read-write}	1.16	open, claim & enable
VideoResolutionList:	<i>string</i>	{read-only}	1.16	open
VideoRecordingType:	<i>string</i>	{read-write}	1.16	open, claim & enable
VideoTypeList:	<i>string</i>	{read-only}	1.16	open

- [Goto Table1-30](#)
- [Goto Table 1-32](#)
- [Goto Table 1-33](#)
- [Goto Table 1-34](#)
- [Goto Table1-35](#)
- [Goto Table 1-36](#)

UPOS Ver1.16 RCSD Specification

Methods (UML operations)

Common

Name	Version
open (logicalDeviceName: <i>string</i>): void {raises-exception}	1.16
close (): void {raises-exception, use after open}	1.16
claim (timeout: <i>int32</i>): void {raises-exception, use after open}	1.16
release (): void {raises-exception, use after open, claim}	1.16
checkHealth (level: <i>int32</i>): void {raises-exception, use after open, enable}	1.16
clearInput (): void { }	1.16 Not supported
clearInputProperties (): void { }	Not supported
clearOutput (): void { }	Not supported
directIO (command: <i>int32</i> , inout data: <i>int32</i> , inout obj: <i>object</i>): void {raises-exception, use after open}	1.16
compareFirmwareVersion (firmwareFileName: <i>string</i> , out result: <i>int32</i>): void {raises-exception, use after open, enable}	1.16
resetStatistics (statisticsBuffer: <i>string</i>): void {raises-exception, use after open, enable}	1.16
retrieveStatistics (inout statisticsBuffer: <i>string</i>): void {raises-exception, use after open, enable}	1.16
updateFirmware (firmwareFileName: <i>string</i>): void {raises-exception, use after open, enable}	1.16
updateStatistics (statisticsBuffer: <i>string</i>): void {raises-exception, use after open, enable}	1.16

Specific

Name	Version
readFrame (frameData: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16
startVideoRecording (fileName: <i>string</i> , overwrite: <i>boolean</i> , recordingTime: <i>int32</i>): void {raises-exception, use after open, claim, enable}	1.16
stopVideoRecording (): void {raises-exception, use after open, claim, enable}	1.16
takePhotograph (fileName: <i>string</i> , overwrite: <i>boolean</i> , int32 , timeout: <i>int32</i>): void {raises-exception, use after open, claim, enable}	ぼお 1

[Goto Table 1-37](#)
[Goto Table 1-38](#)
[Goto Table 1-39](#)
[Goto Table 1-40](#)
[Goto Table 1-41](#)

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

Name	Type	Mutability	Version
upos::events::DataEvent	int32	Not supported (read-only)	
Status:			
upos::events::DirectIOEvent			1.16
EventNumber:	int32	{read-only}	
Data:	int32	{read-write}	
Obj:	object	{read-write}	
upos::events::ErrorEvent			1.16
ErrorCode:	int32	{read-only}	
ErrorCodeExtended:	int32	{read-only}	
ErrorLocus:	int32	{read-only}	
ErrorResponse	int32	{read-write}	
upos::events::OutputCompleteEvent		Not supported	
upos::events::StatusUpdateEvent			1.16
Status:	int32	{read-only}	
upos::events::TransitionEvent		Not supported	1.16

[Goto Table 1-42](#)

[Goto Table 1-43](#)

[Goto Table 1-44](#)

UPOS Ver1.16 RCSD Specification

General Information

The Video Capture Device name is “Video Capture”.

Capabilities

Video capture device class has the following capabilities:

~~— Get the captured frame data.~~

- Take a photo~~graph~~ and record it ~~in a file~~ as a file in a host and may store it in the targeted storage device.
- Take a video~~movie~~ and record it ~~in a file~~ as a file in a host and may store it in the targeted storage device.
- May read~~Read~~ the encoded data from the bar code label with the hydra connected scanner device.
- May detect the individuals faces and/or objects with the hydra connected individual recognition device.

~~— Detect the objects such as faces.~~

[Goto Table 1-45](#)

UPOS Ver1.16 RCSD Specification
Video Capture Class Diagram

The following diagram shows the relationships between the Video Capture classes.

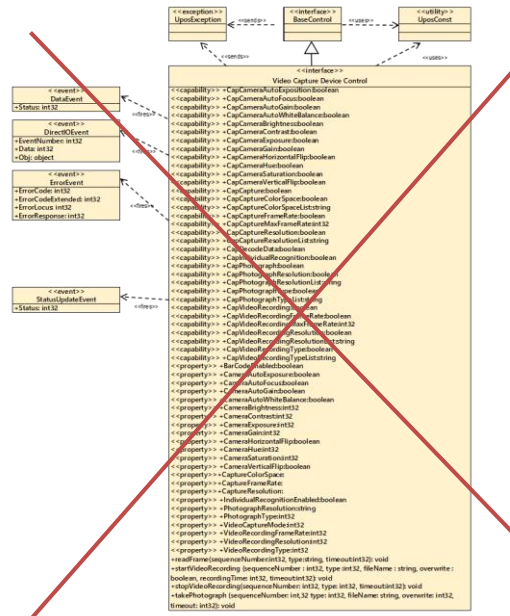
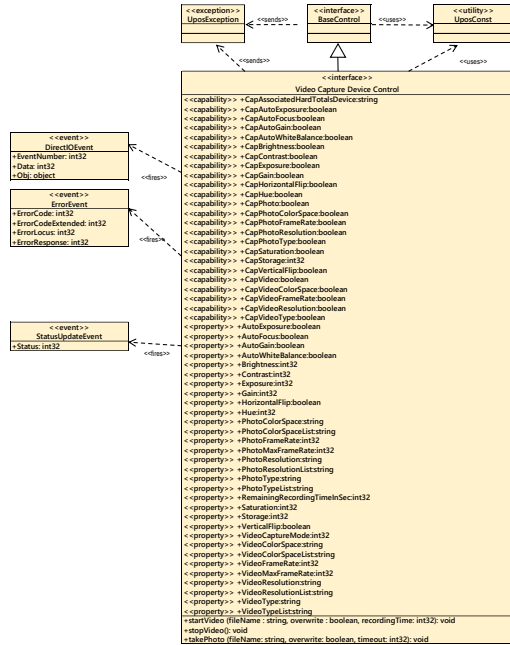


Fig. Chap. 39-1 Video Capture Class Diagram

[Goto Table2-4](#)

UPOS Ver1.16 RCSD Specification Model

Modes

~~When video capture is enabled, the capture begins and the frame data can be retrieved by calling the readFrame method.~~

~~The resolution and frame rate of the frame data to be acquired depend on the operation mode set in the VideoCaptureMode property.~~

~~The following shows the setting to refer to each operation mode and the property for confirming valid values:~~

The Video Capture Device has two operation modes.

- Photo Mode
- Video Mode

The operation of each mode is as follows.

- Photo Mode

Photo Mode may capture a photo image and may save it in a host as the image data file format, if **CapPhoto** property is true. Its' capable data file format is indicated in the **PhotoType** property and all of the capable values are listed in the **PhotoTypeList** property. And the device may save the file in the targeted storage device that is specified by the **Storage** property, if **CapStorage** value is VCAP_CST_HARDTOTALS_ONLY or VCAP_CST_ALL.

- Video Mode

Video Mode may capture a video image data and may save it in a host as the video image data file format, if **CapVideo** property is true. Its' capable data file format is indicated in the **VideoType** property and all of the capable values are listed in the **VideoTypeList** property. And the device may save the file in the targeted storage device that is specified by the **Storage** property, if **CapStorage** value is VCAP_CST_HARDTOTALS_ONLY or VCAP_CST_ALL.

~~Capture-only mode~~

~~Color space: CaptureColorSpace property
=> Valid value confirmation with CapCaptureColorSpaceList property~~

~~Resolution: CaptureResolution property
=> Valid value confirmation with CapCaptureResolutionList property~~

~~Frame rate: CaptureFrameRate property
=> Valid value confirmation with CapCaptureMaxFrameRate property~~

Photo shooting mode

~~Color space: CaptureColorSpace property
=> Valid value confirmation with CapCaptureColorSpaceList property~~

~~Resolution: PhotographResolution property
=> Valid value confirmation with CapPhotographResolutionList property~~

~~Frame rate: CaptureFrameRate property
=> Valid value confirmation with CapCaptureMaxFrameRate property~~

~~Remarks: You can take pictures with takePhotograph method only in this mode.~~

[Goto Table 1-46](#)

UPOS Ver1.16 RCSD Specification

Movie shooting mode

~~Color space: CaptureColorSpace property~~

~~=> Valid value confirmation with CapCaptureColorSpaceList property~~

~~Resolution: VideoRecordingResolution property~~

~~=> Valid value confirmation with the CapVideoRecordingResolutionList property~~

~~Frame rate: VideoRecordingFrameRate property~~

~~=> Valid value confirmation with CapVideoRecordingMaxFrameRate property~~

~~Remarks: It is possible to shoot movies with the startVideoRecording method only in this mode. Since the captured image / movie file is recorded in the area managed by the "hard total" service, the application must also support "hard total" service.~~

Input Model

~~Video capture control follows a common input model of event driven input, although there are some differences:~~

~~"Control" raises a DataEvent event when the recording started by the startVideoRecording method. And it ends when the specified time elapses and the recording to the specified file is completed.~~

~~When an application calls the stopVideoRecording method to end recording, DataEvent event will not occur."~~

~~Also, by activating the FaceCatchEnabled property, face recognition is started, and even when a face is recognized, a DataEvent event is generated.~~

~~To distinguish between Recording Completed to File by Recording and DataEvent event of Face Recognition, refer to the DataEventType property.~~

~~The control sets VCP_ET_VIDEO when recording to the file by recording is completed, and sets VCP_ET_FACECATCH to the DataEventType property when recognizing the face."~~

~~If the AutoDisable property is true, control will be disabled automatically when queuing DataEvent event.~~

~~If the DataEventEnabled property is true, the queued DataEvent is notified to the application. Just before triggering this event, the control copies the data to the property and sets the DataEventEnabled property to false to prevent further data events firing. This allows the control to queue subsequent input data while the application is processing the current input and processing the related properties. When the application finishes processing the current input data and is ready for the next data processing, setting the DataEventEnabled property to true will notify the Data Event again.~~

~~If an error occurs in the control while reading or processing the input data, an ErrorEvent is issued, and if the DataEventEnabled property is true, the application is notified.~~

~~By reading the DataCount property you get the number of Data Events queued by the control.~~

~~All input data queued in the control can be deleted by calling the clearInput method.~~

~~All data properties entered by DataEvent or ErrorEvent occurrence can be restored to the default value by calling the clearInputProperties method.~~

[Goto Table 1-47](#)

Bar Code Scan

~~By setting the BarCodeEnabled property to true for video capture, it is possible to scan the bar code by the camera.~~

~~When reading data from the bar code, the DataEvent event is queued in the scanner service object.~~

UPOS Ver1.16 RCSD Specification

~~Scanned data is stored in the **ScanData** property. If the application sets the **DecodeData** property to true, the data is decoded to **ScanDataLabel** and **ScanDataType**.~~

[Goto Table 1-48](#)

~~Individual Recognition~~

~~By setting the **IndividualRecognitionEnabled** property to true for video capture, it is possible for objects to be recognized by the camera.~~

~~When an object is detected, a **DataEvent** is queued in the object recognition service object.~~

~~The detected data is stored in the **IndividualRecognitionInformation** and **IndividualIDs** of **Individual Recognition Device** properties.~~

[Goto Table 1-49](#)

Device behaviors

“Video capture device” device control follows the device behavior as follows.

They are different in each mode as described below.

Photo Mode

If **CapPhoto** property is true, this mode can be executed.

Prior to start this mode, “**Video Capture Device**” device control needs to set the **VideoCaptureMode** property as to be VCAP_VCMODE_PHOTO. And each of **CapPhotoColorSpace**, **CapPhotoFrameRate**, **CapPhotoResolution**, **CapPhotoType** property is true and these **PhotoColorSpaceList**, **PhotoMaxFrameRate**, **PhotoResolutionList** and **PhotoTypeList** should have the appropriate values to be used as the photo file data in this targeted device. And then it needs to set the appropriate values in the each of **PhotoColorSpace** property, **PhotoFrameRate** property, **PhotoResolution** property and **PhotoType** property.

It starts photo capturing by executing the **takePhoto** method. Then, “**Video Capture Device**” device control may capture a photo image and may save it in a host as an image data file format specified by the value of **PhotoType** property that is listed in the **PhotoTypeList** property. And may store it in the storage device specified by the **Storage** property, if **CapStorage** value is VCAP_CST_HARTTOTALS_ONLY or VCAP_CST_ALL. Then the file name is set by the **takePhoto** method parameter and can deliver the photo data file to the application. If device needs to be able to write the image data file to an associated Hard Totals device, the **CapAssociatedHardTotalsDevice** property holds the open name of the associated Hard Totals device.

This method is executed synchronously. Only one call to **takePhoto** method can be in progress at a time. An attempt to nest video capture device operations will result in an **UPOSException** being thrown.

When it exceeded the specified parameter time out or when photo file generation is finished or when **clearInput** method is executed, the taking photo process will be ended.

StatusUpdateEvent with status VCAP_SUE_START_PHOTO is evoked when **takePhoto** method is executed to notify the application that recording state has started.

When the taking photo is finished, or the specified time out has been exceeded, a **StatusUpdateEvent** with status VCAP_SUE_END_PHOTO is evoked to notify the application that photo taking has been ended.

An **ErrorEvent** event (or events) is enqueued if an error occurs while gathering or processing input.

If **ErrorEvent** response is ER_CONTINUEINPUT, the process of input processing continues. However, as long as the cause of the error is not resolved, the **ErrorEvent** will occur again immediately.

UPOS Ver1.16 RCSD Specification

If **ErrorEvent** is ER_CLEAR, the input processing process is terminated and the taking photo is discarded.

All enqueued input may be deleted by calling **clearInput** method. See the **clearInput** method description for more details.

Video Mode

Prior to start this mode, “**Video Capture Device**” device control needs to set the **VideoCaptureMode** property as to be VCAP_VCMODE_VIDEO. And each of **CapVideoColorSpace**, **CapVideoFrameRate**, **CapVideoResolution** and **CapVideoType** property is true and these **VideoColorSpaceList**, **VideoMaxFrameRate**, **VideoResolutionList** and **VideoTypeList** should have the appropriate values to be used as the video image data file in this targeted device. And then it needs to set the appropriate values in the each of **VideoColorSpace** property, **VideoFrameRate** property, **VideoResolution** property and **VideoType** property.

It starts video image capturing by executing the **startVideo** method. Then “**Video Capture Device**” device control captures a video image and save it in a host with the filename specified value of **VideoType** property that is listed in the **VideoTypeList** property. And may store it in the storage device specified by the **Storage** property, if **CapStorage** value is VCAP_CST_HARTTOTALS_ONLY or VCAP_CST_ALL. And the file name is set by the **startVideo** method parameter and can deliver the video image data file to the application. This method is executed synchronously.

The video capturing ends after the specified time has elapsed or when **stopVideo** method is called or when **clearInput** method is called.

The remaining video capture recording time in seconds can be obtained from the property **RemainingRecordingTimeInSec**.

StatusUpdateEvent with status VCAP_SUE_START_VIDEO is evoked when **startVideo** method is executed to notify the application that taking video has been started.

When the taking video is finished, or the specified time out has been exceeded, a **StatusUpdateEvent** with status VCAP_SUE_END_VIDEO is evoked to notify the application that taking video has been ended.

If the time specified by the **startVideo** method is FOREVER(-1), execution will continue until the **stopVideo** method is called. When **stopVideo** is called, the previous taking video data may be recorded in a host and deliver to the targeted storage device specified by the **Storage** property, if **CapStorage** property value is VCAP_CST_HARTTOTALS_ONLY or VCAP_CST_ALL. And it can be delivered to the application with the specified file name that is set by the **startVideo** method.

Only one call to **startVideo** method can be in progress at a time. An attempt to nest taking video operations will result in an UPOSException being thrown.

If Error occurs during the execution of the **startVideo** method, application may call the **stopVideo** method to terminate the taking video process or cancel the taking video process by calling the **clearInput** method before ending the **ErrorEvent** processing. After this when the **stopVideo** method is called, the video file data until just before the **ErrorEvent** occur is stored to the host and targeted storage device that is specified by the **Storage** property, if **CapStorage** property value is VCAP_CST_HARTTOTALS_ONLY or VCAP_CST_ALL, and can be delivered to the application.

If **ErrorEvent** response is ER_CONTINUEINPUT, the process of input processing continues. However, as long as the cause of the error is not resolved, the **ErrorEvent** will occur again immediately.

If **ErrorEvent** is ER_CLEAR, the input processing process is terminated and the taking video is discarded.

UPOS Ver1.16 RCSD Specification

An **ErrorEvent** event (or events) is enqueued if an error occurs while gathering or processing input.

If there is no error during the execution of **startVideo** method, it is possible to terminate the taking video process and can stop the taking video anytime. When the **stopVideo** method is called, the video data until just before the method is called, may be recorded in the host and targeted storage device that is specified by the **Storage** property if **CapStorage** property is **VCAP_CST_HARTTOTAL_ONLY** or **VCAP_CST_ALL**, and can deliver it to the application.

All enqueued input may be deleted by calling **clearInput** method. See the **clearInput** method description for more details.

[Goto Table 1-50](#)

Device Sharing

Video capture is an exclusive-use device, as follows:

- The application must claim the device before enabling it.
- The application must claim and enable the device before accessing many video capture-specific properties.
- The application must claim and enable the device before calling methods that manipulate the device.
- See the “Summary” table for precise usage prerequisites.

UPOS Ver1.16 RCSD Specification
Properties (UML attributes)

~~BarCodeEnabled Property~~

~~Syntax **BarCodeEnabled: *boolean* {read-write, access after open}**~~

~~Remarks If true, bar code scan is enabled. If false, bar code scan is disabled.
 This property is initialized to false by the **open** method.~~

~~Errors A UposException may be thrown when this property is accessed.
 For further information, see “**Errors**” on page Intro-20.
 Some possible values of the exception’s *ErrorCode* property are:~~

Value	Meaning
E_ILLEGAL	Bar code scanning function is not supported (If it is set true)

~~See also **CapDecodeData** Property [Goto Table 1-51](#)~~

~~CameraAutoExposure Property~~

Syntax ~~CameraAutoExposure: *boolean* {read-write, access after open}~~

Remarks If true, auto exposure of camera is enabled.
~~If false, auto expose of camera is disabled. Otherwise, it is false.~~
 This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed.
 For further information, see “**Errors**” on page Intro-20.
 Some possible values of the exception’s *ErrorCode* property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified. Or it does not support this function.

See also ~~CapCameraAutoExposition~~ Property [Goto Table 1-52](#)

~~CameraAutoFocus Property~~

Syntax ~~CameraAutoFocus: *boolean* {read-write, access after open}~~

Remarks If true, auto focus of camera is enabled.
~~If false auto focus of camera is disabled. Otherwise, it is false.~~
 This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed.
 For further information, see “**Errors**” on page Intro-20.
 Some possible values of the exception’s *ErrorCode* property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified. Or it does not support this function.

See also ~~CapCameraAutoFocus~~ Property [Goto Table 1-53](#)

UPOS Ver1.16 RCSD Specification

~~Camera~~AutoGain Property

Syntax	Camera AutoGain: <i>boolean</i> {read-write, access after open}				
Remarks	If true, auto gain of camera is enabled. Otherwise it is false. If false, auto gain of camera is disabled. When this property is true, it is possible to read the value of Gain property. However, it is not possible to write and change the value of Gain property. If AutoGain property is false, then, it is possible to read, write and change the value of Gain property. This property is initialized by the open method.				
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified. Or it does not support this function.</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	E_ILLEGAL	An invalid value was specified. Or it does not support this function.
<u>Value</u>	<u>Meaning</u>				
E_ILLEGAL	An invalid value was specified. Or it does not support this function.				
See also	Cap Camera AutoGain Property Camera Gain Property				

[Goto Table 1-54](#)

~~Camera~~AutoWhiteBalance Property

Syntax	Camera AutoWhiteBalance: <i>boolean</i> {read-write, access after open}				
Remarks	If true, auto white balance of camera is enabled. Otherwise, it is false. If false, auto white balance of camera is disabled. This property is initialized by the open method.				
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified. Or it does not support this function.</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	E_ILLEGAL	An invalid value was specified. Or it does not support this function.
<u>Value</u>	<u>Meaning</u>				
E_ILLEGAL	An invalid value was specified. Or it does not support this function.				
See also	Cap Camera AutoWhiteBalance Property				

[Goto Table 1-55](#)

~~Camera~~Brightness property

Syntax	Camera Brightness: <i>int32</i> {read-write, access after open}				
Remarks	Indicate the brightness of camera. Valid values range from 0 to 100. This property is initialized by the open method.				
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified. Or it does not support this function.</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	E_ILLEGAL	An invalid value was specified. Or it does not support this function.
<u>Value</u>	<u>Meaning</u>				
E_ILLEGAL	An invalid value was specified. Or it does not support this function.				
See Also	Cap Camera Brightness Property				

[Goto Table 1-56](#)

UPOS Ver1.16 RCSD Specification

CapAssociatedHardTotalsDevice Property

Syntax	CapAssociatedHardTotalsDevice : <i>string</i> {read-write, access after open}
Remarks	Holds the open name of the associated Hard Totals device if the device is able to write to such devices which is the case if CapStorage is either VCAP_CST_ALL or VCAP_CST_HARDTOTALS_ONLY. If CapStorage is VCAP_CST_HOST_ONLY this property value must be the empty string.
Errors	UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See Also	CapStorage Property Goto Table 1-57

CapCameraAutoExposition Property

Syntax	CapCameraAutoExposition : <i>boolean</i> {read-only, access after open}
Remarks	If true, can change the auto exposition of camera can be changed. Otherwise, it is false. If false cannot change the exposition of camera. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See also	CameraAutoExposition Property Goto Table 1-58

CapCameraAutoFocus Property

Syntax	CapCameraAutoFocus : <i>boolean</i> {read-only, access after open}
Remarks	If true, can change the auto focus of camera. Otherwise, it is false. If false, cannot change the auto focus of camera. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See also	CameraAutoFocus Property Goto Table 1-59

CapCameraAutoGain Property

Syntax	CapCameraAutoGain : <i>boolean</i> {read-only, access after open}
Remarks	If true, automatic gain change of the camera is possible. Otherwise, it is false. If false, automatic gain change of camera is not possible. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See also	CameraAutoGain Property Goto Table 1-60

CapCameraAutoWhiteBalance Property

Syntax	CapCameraAutoWhiteBalance : <i>boolean</i> {read-only, access after open}
Remarks	If true, auto white balance of camera is possible. Otherwise, it is false. If false, auto white balance of camera is not possible. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See also	CameraAutoWhiteBalance Property Goto Table 1-61

UPOS Ver1.16 RCSD Specification

Cap~~Camera~~Brightness Property

- Syntax** Cap~~Camera~~Brightness: *boolean* {read-only, access after open}
- Remarks** If true, the brightness of camera can be changed. *Otherwise, it is false.*
~~If false, the brightness of the camera cannot be changed.~~
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed.
For further information, see “**Errors**” on page Intro-20.
- See also** ~~Camera~~Brightness Property [Goto Table 1-62](#)

Cap~~Camera~~Contrast Property

- Syntax** Cap~~Camera~~Contrast: *boolean* {read-only, access after open}
- Remarks** If true, can change the contrast of camera. *Otherwise, it if false.*
~~If false, cannot change the contrast of camera.~~
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed.
For further information, see “**Errors**” on page Intro-20.
- See also** ~~Camera~~Contrast Property [Goto Table 1-63](#)

Cap~~Camera~~Exposure Property

- Syntax** Cap~~Camera~~Exposure: *boolean* {read-only, access after open}
- Remarks** If true, can change the exposure of camera. *Otherwise, it is false.*
~~If false, cannot change the exposure of camera.~~
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed.
For further information, see “**Errors**” on page Intro-20.
- See also** ~~Camera~~Exposure Property [Goto Table 1-64](#)

Cap~~Camera~~Gain Property

- Syntax** Cap~~Camera~~Gain: *boolean* {read-only, access after open}
- Remarks** If true, can change the gain of camera. *Otherwise, it is false.*
~~If false, cannot change the gain of camera.~~
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed.
For further information, see “**Errors**” on page Intro-20.
- See also** ~~Camera~~Gain Property [Goto Table 1-65](#)

Cap~~Camera~~HorizontalFlip Property

- Syntax** Cap~~Camera~~HorizontalFlip: *boolean* {read-only, access after open}
- Remarks** If true, can change the horizontal flip of camera. *Otherwise, it is false.*
~~If false, cannot change the horizontal flip of camera.~~
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed.
For further information, see “**Errors**” on page Intro-20.
- See also** ~~Camera~~HorizontalFlip Property [Goto Table 1-66](#)

UPOS Ver1.16 RCSD Specification
CapCameraHue Property

Syntax CapCameraHue: *boolean* {read-only, access after open}

Remarks If true, the hue of the camera can be changed. Otherwise, it is false.
 If false, hue of the camera cannot be changed.
 This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed.
 For further information, see "Errors" on page Intro-20.

See also CameraHue Property [Goto Table 1-67](#)

~~**CapCapture Property**~~

~~**Syntax** CapCapture: *boolean* {read-only, access after open}~~

~~**Remarks** If true, it supports the capture function and can call the readFrame method and retrieve the frame data.
 If false, it does not support the capture function and cannot retrieve the frame data. This property is initialized by the open method.~~

~~**Errors** A UposException may be thrown when this property is accessed.
 For further information, see "Errors" on page Intro-20.~~

~~**See also** readFrame Method [Goto Table 1-68](#)~~

~~**CapCaptureColorSpace Property**~~

~~**Syntax** CapCaptureColorSpace: *boolean* {read-only, access after open}~~

~~**Remarks** If true, can change the capture color space.
 If false, cannot change the capture color space.
 This property is initialized by the open method.~~

~~**Errors** A UposException may be thrown when this property is accessed.
 For further information, see "Errors" on page Int [Goto Table 1-69](#)~~

~~**CapCaptureColorSpaceList Property**~~

~~**Syntax** CapCaptureColorSpaceList: *string* {read-only, access after open}~~

~~**Remarks** Color space information supported by the device is indicated in a comma-separated list. Each color space information is composed of the following information and is shown in the following order separated by a colon (":").
 This property is initialized by the open method.~~

Parameter	Description
Color space ID	ID for identifying the color space of RGB, YUV-422, etc.
Depth	Number of bits per 1 pixel

~~**Errors** A UposException may be thrown when this property is accessed.
 For further information, see "Errors" on page Intro-20.~~

~~**See also** CaptureColorSpace Property [Goto Table 1-70](#)~~

UPOS Ver1.16 RCSD Specification

CapCaptureFrameRate Property

Syntax CapCaptureFrameRate: *boolean* {read-only, access after open}

Remarks If true, can change the capture frame rate.
If false, cannot change the capture frame rate.
This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.

[Goto Table 1-71](#)

CapCaptureMaxFrameRate Property

Syntax CapCaptureMaxFrameRate: *int32* {read-only, access after open}

Remarks Indicates the maximum frame rate that can be set for the
CaptureFrameRate property.
This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.

See also CaptureFrameRate Property

[Goto Table 1-72](#)

CapCaptureResolution Property

Syntax CapCaptureResolution: *boolean* {read-only, access after open}

Remarks If true, capture resolution is enabled. If
false, capture resolution is disabled.
This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.

See also CaptureResolution Property

CapCaptureResolutionList Property

Syntax CapCaptureResolutionList: *string* {read-only, access after open}

Remarks Indicating the comma-separated list of possible resolutions for the
CaptureResolution property. Resolution is indicated in "horizontal x
height" format. For example, when you support 320x240, 640x480,
640x360, it is the following: "320 x 240, 640 x 480, 640 x 360".
This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.

See also CaptureResolution Property

[Goto Table 1-73](#)

CapDecodeData Property

Syntax CapDecodeData: *boolean* {read-only, access after open}

Remarks If true, the image scanner can read the bar code data.
The scanned bar code data is sent to the scanner service.
This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.

[Goto Table 1-74](#)

UPOS Ver1.16 RCSD Specification

CapIndividualRecognition Property

Syntax	CapIndividualRecognition: <i>boolean</i> {read-only, access after open}
Remarks	If true, individual recognition function is supported. If false, individual recognition function is not supported. If this property is true, individual recognition can be done by setting IndividualRecognitionEnabled property to true. If false, individual recognition cannot be performed. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “Errors” on page Intro-20.
See also	IndividualRecognitionEnabled Property Goto Table 1-75

CapPhotograph Property

Syntax	CapPhotograph: <i>boolean</i> {read-only, access after open}
Remarks	If true, photograph function is supported , it supports the photo function and can take a photo. And to activate the photo mode, the VideoCaptureMode property value needs to set VCAP_VCMODE_PHOTO . If false, photograph function is not supported , it's not supporting the photo function. If true, it is possible taking a photograph by calling the takePhotograph method. If false, it is not possible taking a photograph. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See also	takePhotograph Method, VideoCaptureMode Property Goto Table 1-76

CapPhotoColorSpace Property

Syntax	CapPhotoColorSpace: <i>boolean</i> {read-only, access after open}
Remarks	If true, can handle and change the photo color space. Otherwise, it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20
See also	PhotoColorSpace Property Goto Table 1-77

CapPhotoFrameRate Property

Syntax	CapPhotoFrameRate: <i>boolean</i> {read-only, access after open}
Remarks	If true, can handle and change the capture frame rate. Otherwise, it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See also	PhotoFrameRate Property Goto Table 1-78

UPOS Ver1.16 RCSD Specification

CapPhotographResolution Property

Syntax	CapPhotoResolution: <i>boolean</i> {read-only, access after open}
Remarks	If true, it is possible changing the photograph resolution taking photo resolution is handled and can be changed. If false, it is not possible changing the photograph resolution . Otherwise, it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See also	PhotoResolution Property Goto Table 1-79

CapPhotographResolutionList Property

Syntax **CapPhotographResolutionList:** *string* {read-only, access after open}

Remarks A comma-separated list of possible resolutions for **PhotographResolution** property.
Resolution is indicated by Syntax "Horizontal x Vertical".
For example, when you support 320x240, 640x480, 640x360, it is the following: "320x240,640x480,640x360"
This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed.
For further information, see “**Errors**” on page Intro-20.

See also [PhotographResolution Property](#) [Goto Table 1-80](#)

CapPhotographType Property

Syntax	CapPhotoType: <i>boolean</i> {read-only, access after open}
Remarks	If true, photograph type can be changed photo image format type can be changed. If false, photograph type cannot be changed . Otherwise, it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.

[Goto Table 1-81](#)

CapPhotographTypeList Property

Syntax **CapPhotographTypeList:** *string* {read-only, access after open}

Remarks A comma-separated list of image format values that can be set for the **PhotographType** property.
For example, when supporting BMP and JPEG, it is the following: "BMP, JPEG"

Note: The notation contents may be different depending on the device.
This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed.
For further information, see “**Errors**” on page Intro-20.

See also [PhotographType Property](#) [Goto Table 1-82](#)

UPOS Ver1.16 RCSD Specification

CapCameraSaturation Property

- Syntax** CapCameraSaturation: *boolean* {read-only, access after open}
- Remarks** If true, can change the saturation of camera. ~~If false, cannot change the saturation of camera.~~ Otherwise, it is false.
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed.
For further information, see “**Errors**” on page Intro-20.
- See also** ~~Camera~~ Saturation Property [Goto Table 1-83](#)

CapStorage Property

- Syntax** CapStorage: *int32* {read-only, access after open}
- Remarks** This is an enumeration and announces where the device is able to write the recorded sound data file to.
It holds one of the following values.
- | <u>Value</u> | <u>Meaning</u> |
|--------------------------|---|
| VCAP_CST_HARDTOTALS_ONLY | Only an associate Hard Totals device is supported. |
| VCAP_CST_HOST_ONLY | Only the host’s file system is supported. |
| VCAP_CST_ALL | Both, the associated Hard Totals device and the host’s file system is supported. |
- This property is initialized by the **open** method.
- If a Hard Totals device is supported the Storage, the property value should be VCAP_CST_HARDTOTALS_ONLY or VCAP_CST_ALL, and the property **CapAssociatedHardTotalsDevice** holds the open name of the associated **Hard Totals** device.
- Errors** UposException may be thrown when this property is accessed.
For further information, see “**Errors**” on page Intro-20.
- See Also** Storage Property, CapAssociatedHardTotalsDevice Property

[Goto Table 1-83](#)

CapCameraVerticalFlip Property

- Syntax** CapCameraVerticalFlip: *boolean* {read-only, access after open}
- Remarks** If true, can change the vertical flip of camera. ~~If false, cannot change the vertical flip of camera.~~ Otherwise, it is false.
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed.
For further information, see “**Errors**” on page Intro-20.
- See also** ~~Camera~~ VerticalFlip Property

[Goto Table 1-85](#)

UPOS Ver1.16 RCSD Specification

CapVideoRecording Property

Syntax	CapVideoRecording: <i>boolean</i> {read-only, access after open}
Remarks	If true, video function is supported. If false video recording function is not supported. Otherwise, it is false. If this property is true, movie -taking video and recording can be done by calling the startVideoRecording method. And to activate the video mode, the VideoCaptureMode property value needs to set VCAP_VCMODE_VIDEO. If false, movie taking video and recording cannot be performed. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See also	StartVideoRecording Method, VideoCaptureMode Property

[Goto Table 1-86](#)

CapVideoColorSpace Property

Syntax	CapVideoColorSpace: <i>boolean</i> {read-only, access after open}
Remarks	If true, can change the color space when taking the video. Otherwise, it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20
See also	VideoColorSpace Property

[Goto Table 1-87](#)

CapVideoRecordingFrameRate Property

Syntax	CapVideoRecordingFrameRate : <i>boolean</i> {read-only, access after open}
Remarks	If true, video recording frame rate can be changed, can change the video frame rate from 1 to up to VideoMaxFrameRate property value. If false, video recording frame rate cannot be changed. Otherwise, it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See also	VideoMaxFrameRate Property, VideoFrameRate Property

[Goto Table 1-88](#)

CapVideoRecordingMaxFrameRate Property

Syntax	CapVideoRecordingMaxFrameRate : <i>int32</i> {read-only, access after open}
Remarks	Indicates the maximum frame rate that can be set in VideoRecordingFrameRate property. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See also	VideoRecordingFrameRate Property

[Goto Table 1-89](#)

UPOS Ver1.16 RCSD Specification

CapVideoRecordingResolution Property

Syntax	CapVideoRecordingResolution: <i>boolean</i> {read-only, access after open}
Remarks	If true, video recording resolution taking video resolution can be changed and all of possible values are listed in the VideoResolutionList property values. If false, video recording taking video resolution cannot be changed. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See also	VideoResolutionList Property, VideoResolution Property

[Goto Table 1-90](#)

CapVideoRecordingResolutionList Property

Syntax	CapVideoRecordingResolutionList :string {read-only, access after open}
Remarks	A comma-separated list of possible resolutions for the VideoRecordingResolution property. Resolution is indicated by "Horizontal x Vertical" format. For example, when it supports 320x240, 640x480, 640x360, it is the following: "320x240,640x480,640x360". This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See also	VideoRecordingResolution Property

[Goto Table 1-91](#)

CapVideoRecordingType Property

Syntax	CapVideoRecordingType: <i>boolean</i> {read-only, access after open}
Remarks	If true, video recording taking video type can be changed and all of possible values are listed in the VideoTypeList values. If false, video recording type cannot be changed. Otherwise, it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See also	VideoTypeList Property, VideoType Property

[Goto Table 1-91](#)

CapVideoRecordingTypeList Property

Syntax	CapVideoRecordingTypeList: string {read-only, access after open}
Remarks	A comma-separated list of image format values that can be set for the VideoRecordingType property. For example, when AVI_IYUV, AVI_MJPEG is supported, it is the following: "AVI_IYUV, AVI_MJPEG". Note: The notation contents may be different depending on the device. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See also	VideoRecordingType Property

[Goto Table 1-93](#)

UPOS Ver1.16 RCSD Specification

Camera Contrast Property

Syntax ~~Camera~~ Contrast: *int32* {read-write, access after open}

Remarks Indicate the contrast of the camera. Valid values range from 0 to 100. This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20. Some possible values of the exception’s *ErrorCode* property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified. Or it does not support this function.

See Also Cap~~Camera~~ Contrast Property

[Goto Table 1-94](#)

Camera Exposure Property

Syntax ~~Camera~~ Exposure: *int32* {read-write, access after open}

Remarks Indicate the exposure of camera. Valid values range from 0 to 100. This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20. Some possible values of the exception’s *ErrorCode* property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified. Or it does not support this function.

See also Cap~~Camera~~ Exposure Property

[Goto Table 1-95](#)

Camera Gain Property

Syntax ~~Camera~~ Gain: *int32* {read-write, access after open}

Remarks Indicate the gain of camera. Valid values range from 0 to 100. If **AutoGain** property is true, it is possible to read the value of **Gain** property. However, it is not possible to write and change the value of **Gain** property. If **AutoGain** property is false, then, it is possible to read, write and change the value of **Gain** property. This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20. Some possible values of the exception’s *ErrorCode* property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified. Or it does not support this function.

See also Cap~~Camera~~ Gain Property, [AutoGain Property](#)

[Goto Table 1-96](#)

UPOS Ver1.16 RCSD Specification

CameraHorizontalFlip Property

Syntax	Camera HorizontalFlip: <i>boolean</i> {read-write, access after open}				
Remarks	If true, horizontal flip of camera is enabled and it is possible to reverse the camera captured image horizontally. Otherwise, it is false. If false, horizontal flip of camera is disabled. There is a similar property called VerticalFlip property. However, each VerticalFlip property and HorizontalFlip property value can be set independently. This property is initialized by the open method.				
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified. Or it does not support this function.</td></tr></tbody></table>	Value	Meaning	E_ILLEGAL	An invalid value was specified. Or it does not support this function.
Value	Meaning				
E_ILLEGAL	An invalid value was specified. Or it does not support this function.				
See Also	Cap Camera HorizontalFlip property, VerticalFlip property, CapVerticalFlip property				

[Goto Table 1-97](#)

CameraHue Property

Syntax	Camera Hue: <i>int32</i> {read-write, access after open}				
Remarks	Indicate the hue of camera. Valid values range from 0 to 100. This property is initialized by the open method.				
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified. Or it does not support this function.</td></tr></tbody></table>	Value	Meaning	E_ILLEGAL	An invalid value was specified. Or it does not support this function.
Value	Meaning				
E_ILLEGAL	An invalid value was specified. Or it does not support this function.				
See also	Cap Camera Hue Property Goto Table 1-98				

CapturePhotoColorSpace Property

Syntax	CapturePhoto ColorSpace: <i>string</i> {read-write, access after open}				
Remarks	Indicates the photo color space ID of the frame data to be acquired by the Video Capture Device-readFrame method , if CapPhotoColorSpace property is true. Valid values are one of the values listed in the CapCapturePhotoColorSpaceList property. This property is referred to regardless of which operation mode is set by VideoCaptuerMode property. This property is initialized by the open method.				
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified.</td></tr></tbody></table>	Value	Meaning	E_ILLEGAL	An invalid value was specified.
Value	Meaning				
E_ILLEGAL	An invalid value was specified.				
See also	CapCapturePhotoColorSpaceList Property, VideoCaptureMode property, readFrame Method . CapPhotoColorSpace Property				

[Goto Table 1-99](#)

UPOS Ver1.16 RCSD Specification

PhotoColorSpaceList Property

Syntax PhotoColorSpaceList: *string* {read-only, access after open}

Remarks Photo Color space information supported by the device is indicated in a comma-separated list. Each color space information is composed of the following information and is shown in the following order separated by a colon (":").
This property is initialized by the **open** method.

Parameter	Description
<i>Color space ID</i>	ID for identifying the color space of RGB, YUV 422, etc. And they are indicating like "RGB;YUV422;....."
<i>Depth</i>	Number of bits per 1 pixel

Errors A UposException may be thrown when this property is accessed.
For further information, see "**Errors**" on page Intro-20.

See also PhotoColorSpace Property, VideoCaptureMode Property

[Goto Table 1-100](#)

CapturePhotoFrameRate Property

Syntax ~~CapturePhotoFrameRate: int32~~ {read-write, access after open}

Remarks Indicates the frame rate of frame data to be acquired by the Video Capture Device. ~~readFrame method.~~ Valid values range from 1 to ~~CapCapturePhotoMaxFrameRate~~ property. ~~This property is only referenced when VCP_VMC_CAPTURE is set in VideoCaptureMode property.~~ This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed.
For further information, see "**Errors**" on page Intro-20.

Value	Meaning
E_ILLEGAL	An invalid value was specified.

See also ~~CapCapturePhotoMaxFrameRate~~ Property, PhotoMaxFrameRate Property, ~~readFrame Method,~~ VideoCaptureMode Property,

[Goto Table 1-101](#)

PhotoMaxFrameRate Property

Syntax PhotoMaxFrameRate: *int32* {read-only, access after open}

Remarks Indicates the maximum frame rate that can be set for the PhotoFrameRate property.
This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed.
For further Information, see "**Errors**" on page Intro-20.

See also PhotoFrameRate Property, VideoCaptureMode Property

[Goto Table 1-102](#)

UPOS Ver1.16 RCSD Specification
CapPhotoResolution Property

Syntax	CapPhotoResolution : <i>string</i> {read-write, access after open}				
Remarks	Indicate It shows the resolution of the frame data acquired by the Video Capture Device readFrame method and the photo taken and recorded with the takePhoto method. Valid values are one of those listed in CapCapturePhotoResolutionList property. This property is only referenced when VCP_VCM_CAPTURE is set in VideoCaptureMode property. This property is initialized by the open method.				
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20. Some possible values of the exception's <i>ErrorCode</i> property are: <table border="1" data-bbox="395 728 987 795"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>E_ILLEGAL</td> <td>An invalid value was specified.</td> </tr> </tbody> </table>	Value	Meaning	E_ILLEGAL	An invalid value was specified.
Value	Meaning				
E_ILLEGAL	An invalid value was specified.				
See also	CapCapturePhotoResolutionList Property, readFrame Method VideoCaptureMode Property, takePhoto Method Goto Table 1-103				

PhotographResolution Property

Syntax	PhotographResolution : <i>string</i> {read-write, access after open}				
Remarks	It shows the resolution of the frame data acquired by the readFrame method and the photograph taken with the takePhotograph method. Valid values are one of those listed in CapPhotographResolutionList property. This property is referenced only when VCP_VCM_PHOTO is set in VideoCaptureMode property. This property is initialized by the open method.				
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20. Some possible values of the exception's ErrorCode property are: <table border="1" data-bbox="395 1238 987 1305"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>E_ILLEGAL</td> <td>An invalid value was specified.</td> </tr> </tbody> </table>	Value	Meaning	E_ILLEGAL	An invalid value was specified.
Value	Meaning				
E_ILLEGAL	An invalid value was specified.				
See also	CapPhotographResolutionList Property, VideoCaptureMode Property, readFrame Method , takePhotograph Method				

[Goto Table 1-104](#)

PhotoResolutionList Property

Syntax	PhotoResolutionList: <i>string</i> {read-only, access after open}
Remarks	Indicating the comma-separated list of possible resolutions for the PhotoResolution property. Resolution is indicated in "horizontal x height" format. For example, when you support 320x240, 640x480, 640x360, it is the following: "320x240,640x480,640x360". This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See also	PhotoResolution Property, VideoCaptureMode property

[Goto Table 1-105](#)

UPOS Ver1.16 RCSD Specification

IndividualRecognitionEnabled Property

Syntax	IndividualRecognitionEnabled: <i>boolean</i> {read-write, access after open}				
Remarks	If true individual recognition is enabled. If false, individual recognition is disabled. This property is initialized to false by the open method.				
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20. Some possible values of the exception's ErrorCode property are: <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>Individual recognition function is not supported. (If it is set true)</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	E_ILLEGAL	Individual recognition function is not supported. (If it is set true)
<u>Value</u>	<u>Meaning</u>				
E_ILLEGAL	Individual recognition function is not supported. (If it is set true)				
See also	CapIndividualRecognition Property Goto Table 1-106				

PhotographType Property

Syntax	PhotographType: <i>string</i> {read-write, access after open}				
Remarks	Indicates the image data format of photos taken with the takePhotograph method. Valid values are one of the values listed in the CapPhotographTypeList property. This property is referenced only when VCP_VCM_PHOTO is set in VideoCaptureMode property. This property is initialized by the open method.				
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20. Some possible values of the exception's ErrorCode property are: <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified.</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	E_ILLEGAL	An invalid value was specified.
<u>Value</u>	<u>Meaning</u>				
E_ILLEGAL	An invalid value was specified.				
See also	CapPhotographTypeList Property, takePhotograph Method, VideoCaptureMode Property Goto Table 1-107				

PhotoTypeList Property

Syntax	PhotoTypeList: <i>string</i> {read-only, access after open}
Remarks	A comma-separated list of photo image format values that can be set for the PhotoType property. For example, when supporting BMP and JPEG, it is the following. "BMP,JPEG" Note: The notation contents may be different depending on the device. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See also	PhotoType Property, VideoCaptureMode property Goto Tab 1-108

UPOS Ver1.16 RCSD Specification
RemainingRecordingTimeInSec Property

- Syntax** **RemainingRceordingTimeInSec:**
int32 {read-only, access after open-claim-enable}
- Remarks** This property holds the remaining recording time in seconds if a video recording is ongoing. If no video recording is ongoing its value is 0. When a call to method **startVideo** returns, this property initially holds the time passed as argument *recordingTime* to that call. If this argument value is FOREVER (-1), this property also holds this value unchanged until **stopVideo** method has been called. This property is initialized during device set **DeviceEnabled** method to 0.
- Errors** UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
- See Also** **startVideo** Method, **stopVideo** Method [Goto Table 1-109](#)

CameraSaturation Property

- Syntax** ~~Camera~~Saturation: *int32* {read-write, access after open}
- Remarks** Indicate the saturation of camera. Valid values range from 0 to 100. This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see "**Errors**" on page Intro-20. Some possible values of the exception's *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified. Or it does not support this function.
- See also** Cap~~Camera~~Saturation Property [Goto Table 1-110](#)

UPOS Ver1.16 RCSD Specification
Storage Property

- Syntax** **Storage: *int32* {read-write, access after open-claim-enable}**
- Remarks** This is an enumeration and defines where the device writes the recorded video or photo data file to. Should be set before a call to **startVideo** or **takePhoto** method. It holds one of the following values.
- | <u>Value</u> | <u>Meaning</u> |
|-------------------------|--|
| VCAP_ST_HARDTOTALS | The video or photo data file is written to the associated Hard Totals device. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device. |
| VCAP_ST_HOST | The vide or photo data file is written to the host's file system. |
| VCAP_ST_HOST_HARDTOTALS | The video or photo data file is written to the associated Hard Totals device and host's file system. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device. |
- This property is initialized by the **open** method according to the value hold by **CapStorage**. If **CapStorage** has the value VCAP_CST_ALL, it is initialized to VCAP_ST_HOST_HARDTOTALS.
- Errors** UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
- See Also** **CapStorage** Property [Goto Table 1-111](#)

CameraVerticalFlip Property

- Syntax** ~~Camera~~ **VerticalFlip: *boolean* {read-write, access after open}**
- Remarks** If true, vertical flipping of the video is enabled and it is possible to reverse the video or photo image capturing vertically. Otherwise, it is false. ~~If false, vertical flipping of camera is disabled.~~ There is a similar property called **HorizontalFlip** property and each **VerticalFlip** property and **HorizontalFlip** property value can be set independently. This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see "**Errors**" on page Intro-20. Some possible values of the exception's *ErrorCode* property are:
- | <u>Value</u> | <u>Meaning</u> |
|--------------|---|
| E_ILLEGAL | An invalid value was specified. Or it does not support this function. |
- See also** ~~Cap~~ **CameraVerticalFlip** Property, **HorizontalFlip** Property, **CapHorizontalFlip** Property
- [Goto Table 1-112](#)

UPOS Ver1.16 RCSD Specification VideoCaptureMode Property

Syntax	VideoCaptureMode: <i>int32</i> {read-write, access after open}
Remarks	Indicate the operation mode of video capture device. Valid values are as follows

Parameter	Description
-----------	-------------

~~VCP_VCMODE_CAPTURE~~

~~This mode is for capture only.
The values of the CaptureColorSpace,
CaptureResolution, and CaptureFrameRate
properties are applied to the color space,
resolution, and frame rate of frame data that can
be acquired with the readFrame.~~

VCAP_VCMODE_PHOTO

This mode is for taking photograph and their data recording. Can be set when CapPhoto property is true. The values of the CaptureColorSpace and CaptureFrameRate properties are applied to the color space and frame rate of the frame data that can be acquired by the readFrame method, and the resolution is applied to the resolution of the CapPhotographResolution property. The values of the PhotoType property, PhotoColorSpace property, PhotoResolution property, PhotoFrameRate property are applied to the taking photo image formats list in the PhotoTypeList property, the color space values list in the PhotoColorSpaceList property, the resolution values list in the PhotoResolutionList property, and the frame rate values within the values of PhotoMaxFrameRate property. And taking photo is executed by the takePhoto method.

UPOS Ver1.16 RCSD Specification

VCAP_VCMODE_VIDEO

~~This mode is for capture and movie shooting.~~

This mode is for taking the videos and their data recording. Can be set when **CapVideo** property is true. ~~The value of the **CaptureColorSpace** property is applied to the color space of the frame data that can be acquired by the **readFrame** method, the values of the **CapVideoRecordingResolution** property and the **CapVideoRecordingFrameRate** property are applied to the resolution and the frame rate.~~

The value of the **VideoType** property, **VideoColorSpace** property, **VideoResolution** property and **VideoFrameRate** property are applied to the taking video image format list in the **VideoTypeList** property, the color space values list in the **VideoColorSpaceList** property, the resolution values list in the **VideoResolutionList** property and frame rate values within the values of **VideoMaxFrameRate** property. Taking the videos and their data recording will be executed by the **startVideo** method and ends taking the video by using the **stopVideo** method.

~~This property is initialized to **VCP_VCMODE_CAPTURE** by the **open** method. Indicate the operation mode of video capture.~~

This property is initialized by the by the **open** method. The default value of this property is **VCAP_VCMODE_PHOTO**.

Errors A UposException may be thrown when this property is accessed. For further information, see "**Errors**" on page Intro-20.

See also ~~**CaptureColorSpace** Property, **CaptureResolution** Property, **CaptureFrameRate** Property, **CapPhotographResolution** Property, **CapVideoRecordingResolution** Property, **CapVideoRecordingFrameRate** Property, **readFrame** Method~~
PhotoColorSpace Property, **VideoColorSpace** Property, **PhotoResolution** Property, **VideoResolution** Property, **VideoFrameRate** Property, **PhotoFrameRate** Property, **CapPhotoColorSpace** Property, **CapVideoColorSpace** Property, **CapPhotoResolution** Property, **CapVideoResolution** Property, **VideoMaxFrameRate** Property, **PhotoMaxFrameRate** Property, **VideoCaptureMode** Property, **CapPhoto** Property, **CapVideo** Property, **VideoType** Property, **VideoTypeList** Property **PhotoType** Property, **PhotoTypeList** Property, **takePhoto** Method, **startVideo** Method, **stopVideo** Method.

[Goto Table 1-113](#)

UPOS Ver1.16 RCSD Specification

VideoColorSpace Property

Syntax	VideoColorSpace: <i>string</i> {read-write, access after open}				
Remarks	Indicates the video color space ID of the frame data to be used by startVideo method. Valid values are one of the values listed in the VideoColorSpaceList property. This property is referred to when VideoCaptureMode property value is VCAP_VCMODE_VIDEO and CapVideo is true. This property is initialized by the open method.				
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified.</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	E_ILLEGAL	An invalid value was specified.
<u>Value</u>	<u>Meaning</u>				
E_ILLEGAL	An invalid value was specified.				
See also	VideoColorSpaceList Property, VideoCaptureMode Property, startVideo Method Goto Table 1-114				

VideoColorSpaceList Property

Syntax	VideoColorSpaceList: <i>string</i> {read-only, access after open}						
Remarks	Video Color space information supported by the device is indicated in a comma-separated list. Each color space information is composed of the following information and is shown in the following order separated by a colon (":"). This property is initialized by the open method. <table><thead><tr><th><u>Parameter</u></th><th><u>Description</u></th></tr></thead><tbody><tr><td><i>Color space ID</i></td><td>ID for identifying the color space of RGB, YUV 422, etc. And they are indicating like“RGB;YUV422;.....”</td></tr><tr><td><i>Depth</i></td><td>Number of bits per 1 pixel</td></tr></tbody></table>	<u>Parameter</u>	<u>Description</u>	<i>Color space ID</i>	ID for identifying the color space of RGB, YUV 422, etc. And they are indicating like“RGB;YUV422;.....”	<i>Depth</i>	Number of bits per 1 pixel
<u>Parameter</u>	<u>Description</u>						
<i>Color space ID</i>	ID for identifying the color space of RGB, YUV 422, etc. And they are indicating like“RGB;YUV422;.....”						
<i>Depth</i>	Number of bits per 1 pixel						
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.						
See also	CapVideoColorSpace Property, VideoColorSpace Property						

[Goto Table 1-115](#)

UPOS Ver1.16 RCSD Specification

VideoRecordingFrameRate Property

Syntax	VideoRecordingFrameRate; <i>int32</i> {read-write, access after open}				
Remarks	Indicates the frame rate of the frame data acquired recorded by the Video Capture Device readFrame method and the movie taken video image capturing and recorded with the startVideoRecording method. Valid values range from 1 to VideoMaxFrameRate property and CapVideo property is true. This property is only applied when VCAP_VCMODE_VIDEO is set in VideoCaptureMode property. This property is only referred when VCP_VCM_VIDEO is set in VideoCaptureMode property. This property is initialized by the open method.				
Errors	A UposException may be thrown when this property is accessed. For further information, see “Errors” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified.</td></tr></tbody></table>	Value	Meaning	E_ILLEGAL	An invalid value was specified.
Value	Meaning				
E_ILLEGAL	An invalid value was specified.				
See also	CapVideoRecordingMaxFrameRate Property, VideoCaptureMode Property, readFrame Method , startVideoRecording Method CapVideo Property Goto Table 1-116				

VideoMaxFrameRate Property

Syntax	VideoMaxFrameRate; <i>int32</i> {read-only, access after open}
Remarks	Indicates the maximum video recording frame rate that can be set in VideoFrameRate property. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “Errors” on page Intro-20.
See also	VideoFrameRate Property Goto Table 1-117

VideoRecordingResolution Property

Syntax	VideoRecordingResolution; <i>string</i> {read-write, access after open}				
Remarks	Indicates the resolution of video image data the frame data acquired by the readFrame method Video Capture Device and the photograph taken with the recorded with the execution of startVideoRecording method. Valid values are one of the values listed in the CapVideoRecordingResolutionList property. This property is only applied when VCAP_VCMODE_VIDEO is set in VideoCaptureMode property and if CapVideo property is true. This property is only referred when VCP_VCM_VIDEO is set in VideoCaptureMode property. This property is initialized by the open method.				
Errors	A UposException may be thrown when this property is accessed. For further information, see “Errors” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified.</td></tr></tbody></table>	Value	Meaning	E_ILLEGAL	An invalid value was specified.
Value	Meaning				
E_ILLEGAL	An invalid value was specified.				
See also	CapVideoRecordingResolutionList Property, CapVideo Property, VideoCaptureMode Property, readFrame Method , startVideoRecording Method Goto Table 1-118				

UPOS Ver1.16 RCSD Specification

VideoResolutionList Property

Syntax	VideoResolutionList : <i>string</i> {read-only, access after open}
Remarks	A comma-separated list of possible resolutions for the VideoResolution property. Resolution is indicated by "Horizontal resolution number x Vertical resolution number" format. For example, when it supports 320x240, 640x480, 640x360, it is the following: "320x240,640x480,640x360" This property is initialized by the open method.
Errors	A UpoException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See also	CapVideoResolution Property, VideoResolution Property

[Goto Table 1-119](#)

VideoRecordingType Property

Syntax	VideoRecordingType ; <i>string</i> {read-write, access after open}				
Remarks	Indicate the shape of the taking video movie taken and recorded with the startVideoRecording method. Valid values are one of those listed in CapVideoRecordingTypeList property. This property is only referred when VCP_VCM_VIDEO is set in VideoCaptureMode property. This property is applied when VCAP_VCMODE_VIDEO is set in VideoCaptureMode property and if CapVideo property is true. This property is initialized by the open method.				
Errors	A UpoException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20. Some possible values of the exception's <i>ErrorCode</i> property are: <table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified.</td></tr></tbody></table>	Value	Meaning	E_ILLEGAL	An invalid value was specified.
Value	Meaning				
E_ILLEGAL	An invalid value was specified.				
See also	VideoCaptureMode Property, CapVideo Property, CapVideoRecordingTypeList Property, startVideoRecording Method				

[Goto Table 1-120](#)

UPOS Ver1.16 RCSD Specification

VideoTypeList Property

Syntax	VideoTypeList: <i>string</i> {read-only, access after open}
Remarks	A comma-separated list of image format values that can be set for the VideoType property. *1For example, when AVI_IYUV, AVI_MJPEG is supported, it is the following "AVI_IYUV,AVI_MJPEG". Note: The notation contents may be different depending on the device. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See also	CapVideoType Property, VideoType Property

[Goto Table 1-121](#)

Note *1: The Video type related information are listed in here as the reference.

AVI : Digital container format :
https://en.wikipedia.org/wiki/Digital_container_format

MJPEG : Motion JPEG :
https://en.wikipedia.org/wiki/Motion_JPEG

IYUV : 4:2:0 Video Pixel Formats :
<https://docs.microsoft.com/en-us/windows-hardware/drivers/display/4-2-0-video-pixel-formats>

4:2:2 Video Pixel Formats :
<https://docs.microsoft.com/en-us/windows-hardware/drivers/display/4-2-2-video-pixel-formats>

Video Formats and their Abbreviation :
<http://technewzbd.blogspot.com/2013/05/video-formats-and-their-abbreviation.html>

Note: Video Capture Device Property Value Relationship

Properties listed below are related within each Photo / Video Mode group, and if any value change occurs, other values may change accordingly.

Photo Mode Group Properties

PhotoType, PhotoColorSpace, PhotoColorSpaceList, PhotoFrameRate, PhotoMaxFrameRate, PhotoResolution, PhotoResolutionList

Video Mode Group Properties

VideoType, VideoColorSpace, VideoColorSpaceList, VideoFrameRate, VideoMaxFrameRate, VideoResolution, VideoResolutionList

[Goto Table 1-122](#)

UPOS Ver1.16 RCSD Specification

Methods (UML operations)

~~readFrame Method~~

Syntax	readFrame (frameData: string): void (raises exception, use after open, claim, enable)
	Parameter Description
	frameData Indicates the area where frame data is stored.
Remarks	Acquires the captured frame data and stores it in frameData. The color space and resolution of frame data differs depending on the operation mode set in the VideoCaptureMode property. For details, refer to the VideoCaptureMode property. This method is executed synchronously.
Errors	A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20. Some possible values of the exception's <i>ErrorCode</i> property are:
	Value Meaning
	E_ILLEGAL This function is not supported
See also	VideoCaptureMode Property Goto Teble 1-123

UPOS Ver1.16 RCSD Specification
startVideoRecording Method

Syntax **startVideoRecording** (fileName : string, overwrite: *boolean*, recordingTime: *int32*): void{raises-exception, use after open-claim-enable}

Parameter	Description
filename	Specify the name of the movie video file to be recorded.
Overwrite	Specify the behavior when the same name file exists. If true, it is overwritten. If false, it will raise the UposException.
recordingTime	Specify the time for video recording in seconds. If FOREVER (-1) is specified, recording will continue until the stopVideo method is called.

Remarks ~~Recording starts with the setting contents of the CaptureColorSpace and VideoRecordingResolution properties, and recording starts in the format set by the VideoRecordingType property.~~ Before calling this method, it needs to set the **VideoCaptureMode** property to VCAP_VCMODE_VIDEO and **CapVideo** property needs to be true. Video capturing and recording starts with the setting contents of the **VideoColorSpace** property, **VideoResolution** property, **VideoFrameRate** property and **VideoType** property. This method is executed synchronously. **StatusUpdateEvent** will notify the application that there is a change in the power status or a state change during video capturing and recording. When the time specified in recordingTime has elapsed, or by calling the **stopVideoRecording** method, recording is completed and the ~~movie~~ video file specified by fileName is recorded and can deliver to the application. Also, S_BUSY is set in the **Status** property during ~~movie execution~~ video capturing and recording. The place where video files are recorded is ~~the area managed by "Hard Total" service~~ controlled through the **Storage** Property.

Errors A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20. Some possible values of the exception's *ErrorCode* property are:

Value	Meaning
E_ILLEGAL	fileName is too long or contains characters that cannot be used, or 0 is specified for recordingTime.
	VideoCaptureMode property is not VCAP_VCMODE_VIDEO
	E_EXISTS fileName already exists. (If overwrite is false)
	E_BUSY Cannot execute because it is recording.

See also ~~Capture~~**VideoColorSpace** Property, ~~VideoRecording~~**Resolution** Property, **VideoFrameRate** Property, ~~VideoRecording~~**Type** Property, ~~stopVideoRecording~~ Method, **StatusUpdateEvent** Event, **VideoCaptureMode** Property

[Goto Table 1-124](#)

UPOS Ver1.16 RCSD Specification
stopVideoRecording Method

Syntax stopVideoRecording ():
 void {raises-exception, use after open-claim-enable}

Remarks The video capturing and recording process started by the startVideoRecording method has been ended and the taking video recording of the movie image file is completed. This method processed synchronously. StatusUpdateEvent will notify the application that there is a change in the power status or a state change during taking video and recording.

Errors A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20. Some possible values of the exception's ErrorCode property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	It is not recorded.

See also startVideoRecording Method, StatusUpdateEvent Event
[Goto Table 1-124](#)

UPOS Ver1.16 RCSD Specification
takePhotograph Method

Syntax **takePhotograph** (fileName: *string*,
 overwrite: *boolean*, timeout: *int32*):
 void{raises-exception, use after open-claim-enable}

Parameter	Description
fileName	Specify the image file name to be recorded.
overwrite	Specify the behavior when the same name file exists. If true it overwrites. If false, UposException is thrown.
timeout	Allowed execution time in milliseconds, before the method fails and a timeout ErrorEvent is sent to the application. If FOREVER (-1) the service will wait until a photograph is taken or an application error occurs.

Remarks Take photo and record with setting contents of **CapturePhotoColorSpace** property, **PhotographResolution** property, **PhotoFrameRate** Property and **PhotographType** property ~~and record images~~. Before calling this method, it needs to set the **VideoCaptureMode** property to VCAP_VCMODE_PHOTO and ~~change to the photo shooting mode. this method can be executed if CapPhoto property is true.~~ This method is executed synchronously. The location where image photo files are recorded is ~~the area managed by "Hard Total" service controlled through the Storage Property.~~ The timeout specifies the number of milliseconds

Errors A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20. Some possible values of the exception's *ErrorCode* property are:

Value	Meaning
E_ILLEGAL	One of the following occurred. FileName is too long or contains unusable characters. VideoCaptureMode property is not VCAP_VCMODE_PHOTO and CapPhoto property is not true.
E_EXISTS	fileName already exist. (When overwrite=false)

See also **VideoCaptureMode** Property, **CapturePhotoColorSpace** Property, **PhotographResolution** Property, **CapPhoto** Property, **PhotographType** Property, **PhotoFrameRate** Property, **StatusUpdateEvent** Event

[Goto Table 1-126](#)

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

~~DataEvent~~

~~<<event>> upos::events::DataEvent
 Status: *int32* {read-only}~~

~~**Description** Notifies the application when data from the Video Capture device is available to be read.~~

~~**Attributes** This event contains the following attributes:~~

Attribute	Type	Description
Status	int32	Set to 0.

~~**Remarks** Before this event is delivered, the Video Capture movie image is placed into readFrame.~~

~~This event is to be used only for those types of vendor specific functions that are not otherwise described.~~

~~Use of this event may restrict the application program programform being used with other vendor's devices which may not have any knowledge of the Service's need for this event.~~

~~**See Also** "Events" on page Intro-19, directIO method [Goto Table 1-127](#)~~

DirectIOEvent

<<event>> upos::events::DirectIOEvent
 EventNumber : *int32* {read-only}
 Data : *int32* {read-write}
 Obj : *object* {read-write}

Description Provides Service information directly to the application. This event provides a means for a vendor-specific Video Capture Service to provide events to the application that are not otherwise supported by the [device](#) control.

Attributes This event contains the following attributes:

Attribute	Type	Description
EventNumber	int32	Event number whose specific values are assigned by the Service.
Data	int32	Additional numeric data. Specific values vary by the EventNumber and the Service. This attribute is settable.
Obj	object	Additional data whose usage varies by the EventNumber and the Service. This attribute is settable.

Remarks This event is to be used only for those types of vendor specific functions that are not otherwise described.

Use of this event may restrict the application program programform being used with other vendor's devices which may not have any knowledge of the Service's need for this event.

See Also "Events" on page Intro-19, directIO method [Goto Table 1-128](#)

UPOS Ver1.16 RCSD Specification
ErrorEvent

```
<<event>> upos::events::ErrorEvent
  ErrorCode      : int32 {read-only}
  ErrorCodeExtended : int32 {read-only}
  ErrorLocus     : int32 {read-only}
  ErrorResponse  : int32 {read-write}
```

Description Notifies the application that a Video Capture Device error has been detected and suitable response by the application is necessary to process the error condition.

Attributes This event contains the following attributes:

Attributes	Type	Description
<i>ErrorCode</i>	<i>int32</i>	Error code causing the error event. See a list of Error Codes on page 20.
<i>ErrorCodeExtended</i>	<i>int32</i>	Extended Error code causing the error event. If <i>ErrorCode</i> is E_EXTENDED, then see values below. Otherwise, it may contain a Service-specific value.
<i>ErrorLocus</i>	<i>int32</i>	Location of the error. If EL_OUTPUT is specified. An error occurred during asynchronous action.
<i>ErrorResponse</i>	<i>int32</i>	Pointer to the error event response. See ErrorResponse below for values. Error Response, whose default value may be overridden by the application. (i.e., this attribute is settable). See <i>ErrorResponse</i> below for values.

If *ErrorCode* is E_EXTENDED, then *ErrorCodeExtended* has one of the following values:

Value	Meaning
EVCAP_NOROOM	The image data storage area does not have enough room to store.

The *ErrorLocus* attribute has one of the following values:

Value	Meaning
EL_OUTPUT	Error occurred while processing asynchronous output.
EL_INPUT	Error occurred while gathering or processing event-driven input. No previously buffered input data is available.
EL_INPUT_DATA	Error occurred while gathering or processing event-driven input, and some previously buffered data is available.

UPOS Ver1.16 RCSD Specification

The application's error event handler can set the *ErrorResponse* attribute to one of the following values:

<u>Value</u>	<u>Meaning</u>
ER_RETRY	Retry sending the data. The error state is exited. May be valid for some input devices when the locus is EL_INPUT, in which case the input is retried and the error state is exited. Typically, valid for asynchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is retried and the error state is exited. This is the default response when the locus is EL_OUTPUT.
ER_CLEAR	Valid for loci: EL_OUTPUT, EL_INPUT_DATA and EL_OUTPUT. Clear all buffered input or output data (including all asynchronous output). The error state is exited. This is the default response when the locus is EL_INPUT.
ER_CONTINUEINPUT	Only valid when the locus is EL_INPUT_DATA. Acknowledges that a data error has occurred and directs the Device to continue input processing. The Device remains in the error state and will deliver additional DataEvents as directed by the DataEventEnabled property. When all input has been delivered and DataEventEnabled is again set to true, then another ErrorEvent is delivered with locus EL_INPUT. This is the default response when the locus is EL_INPUT_DATA.

- Remarks** This event is enqueued when an error is detected and the Device's **State** transitions into the error state. ~~Input error events are not delivered until DataEventEnabled is true, so that proper application sequencing occurs.~~
- ~~Unlike a DataEvent, the Device does not disable further DataEvents or input ErrorEvents; it leaves the DataEventEnabled property value at true. Note that the application may set DataEventEnabled to false within its event handler if subsequent input events need to be disabled for a period of time.~~
- See Also** ~~"Device Input Model" on page Intro-22, "Error Handling" on page Intro-23, "Device Output Models" on page Intro-25.~~

UPOS Ver1.16 RCSD Specification
StatusUpdateEvent

<< event >>	upos::events::StatusUpdateEvent Status : int32 {read-only}																
Description	<i>Notifies the application that there is a change in the power status or a state change of the Video Capture device.</i>																
Attributes	This event contains the following attribute: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Attributes</th> <th style="text-align: left;">Type</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td><i>Status</i></td> <td><i>int32</i></td> <td>Indicates a change in the power status or a state change of the unit.</td> </tr> </tbody> </table> <p><i>Note that Release 1.3 added Power State Reporting with additional Power reporting StatusUpdateEvent values.</i></p> <p>The Update Firmware capability added additional Status values for communicating the status/progress of an asynchronous update firmware process. See “StatusUpdateEvent” description on page 1-34.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Value</th> <th style="text-align: left;">Meaning</th> </tr> </thead> <tbody> <tr> <td>VCAP_SUE_START_VIDEO_RECORDING</td> <td>It will be notified when video recording starts.</td> </tr> <tr> <td>VCAP_SUE_STOP_VIDEO_RECORDING</td> <td>It will be notified when video recording stops.</td> </tr> <tr> <td>VCAP_SUE_START_PHOTO</td> <td>It will be notified when photo capturing starts.</td> </tr> <tr> <td>VCAP_SUE_END_PHOTO</td> <td>It will be notified when photo capturing ends.</td> </tr> </tbody> </table>	Attributes	Type	Description	<i>Status</i>	<i>int32</i>	Indicates a change in the power status or a state change of the unit.	Value	Meaning	VCAP_SUE_START_VIDEO_RECORDING	It will be notified when video recording starts.	VCAP_SUE_STOP_VIDEO_RECORDING	It will be notified when video recording stops.	VCAP_SUE_START_PHOTO	It will be notified when photo capturing starts.	VCAP_SUE_END_PHOTO	It will be notified when photo capturing ends.
Attributes	Type	Description															
<i>Status</i>	<i>int32</i>	Indicates a change in the power status or a state change of the unit.															
Value	Meaning																
VCAP_SUE_START_VIDEO_RECORDING	It will be notified when video recording starts.																
VCAP_SUE_STOP_VIDEO_RECORDING	It will be notified when video recording stops.																
VCAP_SUE_START_PHOTO	It will be notified when photo capturing starts.																
VCAP_SUE_END_PHOTO	It will be notified when photo capturing ends.																
Remarks	Enqueued when the Video Capture Device detects a power state change or a status change.																
See Also	“Events” on page Intro-19. Goto Table 1-129																

Individual Recognition

This Chapter defines the Individual Recognition device category.

Summary

Properties (UML attributes)

<i>Common</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
AutoDisable:	<i>boolean</i>	{read-write}	1.16	Open
CapCompareFirmwareVersion:	<i>boolean</i>	{read-only}	1.16	Open
CapPowerReporting:	<i>int32</i>	{read-only}	1.16	open
CapStatisticsReporting:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateFirmware:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateStatistics:	<i>boolean</i>	{read-only}	1.16	open
CheckHealthText:	<i>string</i>	{read-only}	1.16	open
Claimed:	<i>boolean</i>	{read-only}	1.16	open
DataCount:	<i>int32</i>	{read-only}	1.16	open
DataEventEnabled:	<i>boolean</i>	{read-write}	1.16	open
DeviceEnabled:	<i>boolean</i>	{read-write}	1.16	open & claim
FreezeEvents:	<i>boolean</i>	{read-write}	1.16	open
OutputID:	<i>int32</i>	{read-only}	1.16	Not Supported
PowerNotify:	<i>int32</i>	{read-write}	1.16	open
PowerState:	<i>int32</i>	{read-only}	1.16	open
State:	<i>int32</i>	{read-only}	1.16	--
DeviceControlDescription:	<i>string</i>	{read-only}	1.16	--
DeviceControlVersion:	<i>int32</i>	{read-only}	1.16	--
DeviceServiceDescription:	<i>string</i>	{read-only}	1.16	open
DeviceServiceVersion:	<i>int32</i>	{read-only}	1.16	open
PhysicalDeviceDescription:	<i>string</i>	{read-only}	1.16	open
PhysicalDeviceName:	<i>string</i>	{read-only}	1.16	open

UPOS Ver1.16 RCSD Specification

Properties (Continued)

<i>Specific</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
CapIndividualList:	<i>string</i>	{read-only}	1.16	open
IndividualIDs:	<i>string</i>	{read-only}	1.16	open, claim & enable
IndividualRecognitionFilter:	<i>string</i>	{read-write}	1.16	open
IndividualRecognitionInformation	<i>string</i>	{read-only}	1.16	open

Methods (UML operations)

Common

<i>Name</i>	<i>Version</i>
open (logicalDeviceName: <i>string</i>): void {raises-exception}	1.16
close (): void {raises-exception, use after open}	1.16
claim (timeout: <i>int32</i>): void {raises-exception, use after open}	1.16
release (): void {raises-exception, use after open, claim}	1.16
checkHealth (level: <i>int32</i>): void {raises-exception, use after open, enable}	1.16
clearInput (): void { }	1.16 Not supported
clearInputProperties (): void { }	1.16 Not supported
clearOutput (): void { }	Not supported
compareFirmwareVersion (firmwareFileName: <i>string</i> , out result: <i>int32</i>): void {raises-exception, use after open, enable}	1.16
directIO (command: <i>int32</i> , inout data: <i>int32</i> , inout obj: <i>object</i>): void {raises-exception, use after open}	1.16
resetStatistics (statisticsBuffer: <i>string</i>): void {raises-exception, use after open, enable}	1.16
retrieveStatistics (inout statisticsBuffer: <i>string</i>): void {raises-exception, use after open, enable}	1.16
updateFirmware (firmwareFileName: <i>string</i>): void {raises-exception, use after open, enable}	1.16
updateStatistics (statisticsBuffer: <i>string</i>): void {raises-exception, use after open, enable}	1.16

[Goto Table 1-130](#)

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

<i>Name</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>
upos::events::DataEvent			1.16
Status:	<i>int32</i>	{read-only}	
upos::events::DirectIOEvent			1.16
EventNumber:	<i>int32</i>	{read-only}	
Data:	<i>int32</i>	{read-write}	
Obj:	<i>object</i>	{read-write}	
upos::events::ErrorEvent			1.16
ErrorCode:	<i>int32</i>	{read-only}	
ErrorCodeExtended:	<i>int32</i>	{read-only}	
ErrorLocus:	<i>int32</i>	{read-only}	
ErrorResponse:	<i>int32</i>	{read-write}	
upos::events::OutputCompleteEvent		<i>Not supported</i>	1.16
upos::events::StatusUpdateEvent			1.16
Status:	<i>int32</i>	{read-only}	
upos::events::TransitionEvent		<i>Not supported</i>	1.16

[Goto Table 1-131](#)

UPOS Ver1.16 RCSD Specification General Information

The Individual Recognition programmatic name is “Individual Recognition”.

Capabilities

The Individual Recognition has the following set of capabilities:

Analyzes the image of the camera and recognizes individuals such as people and listed goods.

Individual Recognition Class Diagram

The following diagram shows the relationships between the Individual Recognition classes.

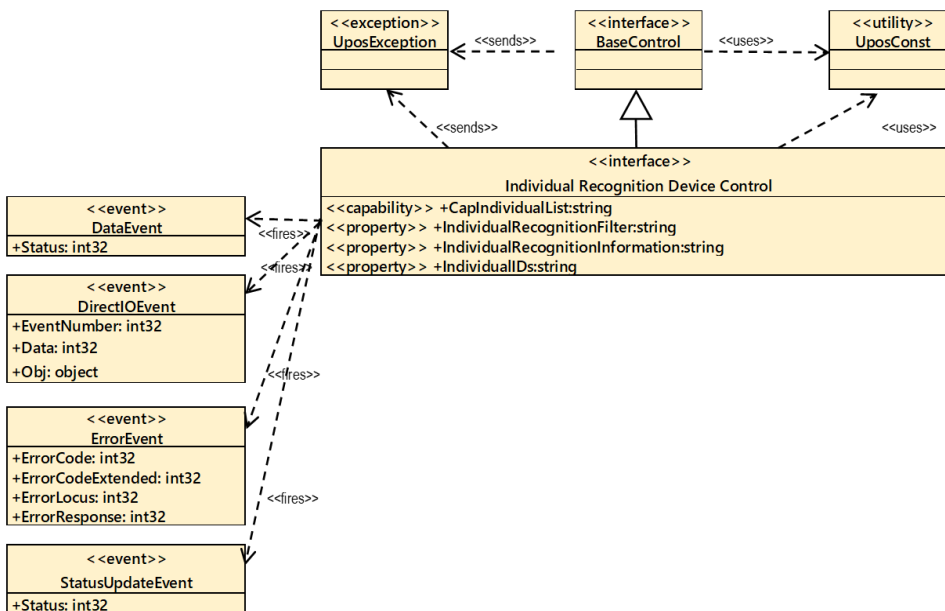


Fig. Chap.40-1 Individual Recognition Class Diagram

UPOS Ver1.16 RCSD Specification

Model

The Individual Recognition follows the general "Device Input Model" for event-driven input:

Input Model

~~The readValue method follows the UnifiedPOS Input model.~~

- When an individual is recognized by this device, a **DataEvent** is delivered to the application after the **IndividualIDs** property was set to indicate the recognized individuals.
 - Identifiable individuals are indicated by the **CapIndividualList** property.
 - Check the functions supported by the device, set validity / invalidity, etc. with the **IndividualRecognitionInformation** property.
 - Recognized data is stored in the **IndividualRecognitionInformation** property, **IndividualIDs** property.
 - How to recognize the individuals depends on the **IndividualRecognitionFilter** function, therefore, please refer to the **IndividualRecognitionFilter** section.
 - Other device behavior about this device supports the general device input model as listed below.
 - If the **AutoDisable** property is true, then the device automatically disables itself when a **DataEvent** is enqueued.
 - An enqueued **DataEvent** can be delivered to the application when the **DataEventEnabled** property is true and other event delivery requirements are met. Just before delivering this event, data is copied into corresponding properties, and further data events are disabled by setting **DataEventEnabled** to false. This causes subsequent input data to be enqueued while the application processes the current input and associated properties. When the application has finished processing the current input and is ready for more data, it reenables events by setting **DataEventEnabled** to true.
 - An **ErrorEvent** (or events) is enqueued if an error occurs while gathering or processing input, and is delivered to the application when **DataEventEnabled** is true and other event delivery requirements are met.
 - The **DataCount** property may be read to obtain the total number of enqueued **DataEvents**.
 - All enqueued input may be deleted by calling **clearInput** method. See the **clearInput** method description for more details.
 - All data properties that are populated as a result of firing a **DataEvent** or **ErrorEvent** can be set back to their default values by calling the **clearInputProperties** method.
 - ~~• Identifiable individuals are indicated by the **CapIndividualList** property.~~
- ~~Check the functions supported by the device, set validity / invalidity, etc. with the **IndividualRecognitionInformation** property.~~
- ~~Recognized data is stored in the **IndividualRecognitionInformation** property, **IndividualIDs**.~~
- The application will be informed about any status change with a **StatusUpdateEvent**, also all corresponding status properties will be updated before event delivery.

[Goto Table 1-132](#)

UPOS Ver1.16 RCSD Specification

Device Sharing

The Individual Recognition is an exclusive-use device, as follows:

- The application must claim the device before enabling it.
- The application must claim and enable the device before the device begins reading input.
- See the “Summary” table for precise usage prerequisites.

UPOS Ver1.16 RCSD Specification

IndividualRecognitionFilter

The **IndividualRecognitionFilter** property defines the following data as information for the individual recognition function of Individual Recognition Device.

- Various support function existence or not.
(Supported functions are defined by the device)
- Enable, disable status of various functions.
- Types handled by various functions (examples: “male”, “female” of gender recognition)
- Filter setting of various functions.

The following data is defined in the IndividualRecognitionInformation property

- Individual Recognition input data

The device defines the individual recognition function information and the individual recognition input data.

The application refers to these contents to determine the support range and so on. In addition, the application changes the enabled / disabled state of various functions, the filter setting, and controls each function.

The enabled / disabled state of the various functions set by the application, and the filter settings are applied by setting the DeviceEnabled property to true and enabling the individual recognition function.

When the application set various functions, it is possible to specify and set only the target ones.

The device fires a DataEvent based on the content set by the application and stores the input data in **IndividualRecognitionInformation** property.

[Goto Table 1-133](#)

UPOS Ver1.16 RCSD Specification

IndividualRecognitionFilter Property Example Format

The IndividualRecognitionFilter property of the individual recognition device may define various information. Here is the example described by using the JSON format.

■ Basic Items

Key	Value	Value change capability	Explanation
IndividualRecognitionFilter	object	N	Information for the various individual recognition. Target device define the supporting individual recognition object.
[IndividualRecognitionID]	object	N	Recognizable individual recognition information. Key name is the ID of recognized individual
Enabled	boolean	Y	Enable or disable state of target individual recognition. Application can control the target individual recognition by referring or changing.
Properties	object	N	Property information of the target individual recognition. Application control the target individual recognition by referring or changing the defined property value.
[Property01]	-	-	
[Property02]	-	-	
Filters	object	N	Input data filter setting information. Application filter the target individual recognition input data by changing the defined value.
[Filter01]	-	-	
[Filter02]	-	-	

[Goto Table 1-133](#)

UPOS Ver1.16 RCSD Specification

■ Face Recognition device example

Key	Value	Value change capability	Explanation
IndividualRecognitionFilter	object	N	
Face	object	N	
Enabled	boolean	Y	
Properties	object	N	
FaceImageNamePrefix	string	Y	Output image file prefix for face recognition
Gender	object	N	Information on gender recognition
Enabled	boolean	Y	Gender recognition enable, disable state
CapTypeList	array	N	Type list ("female", "male")
Age	object	N	Information on age recognition.
Enabled	boolean	Y	Age recognition enable, disable state
Facial Expression	object	N	Information on facial expression recognition
Enabled	boolean	Y	Facial expression recognition enable, disable state.
CapTypeList	array	N	Type list ("smile", "angry",...)
Gaze	object	N	Information on gaze recognition
Enabled	boolean	Y	Gaze recognition enable, disable state.
CapTypeList	array	N	Type list ("gaze", "nogaze")
Distance	object	N	Information on distance recognition
Enabled	boolean	Y	Distance recognition enable, disable state
CapTypeList	array	N	Type list ("near", "far", "very far",...)
NearLength	number	Y	Distance to recognize as "near". A recognition event is fired when a person is recognized in the range from 0 to Near Length.
FarLength	number	Y	Distance to recognize as "far", "very far". A recognition event is fired when a person is recognized in the range from Near Length to Far Length. A recognition event is fired when a person is recognized in the range more than Far Length.
Authentication	object	N	Information on face authentication
Enabled	boolean	Y	Face authentication enable, disable state.
ImageList	array	Y	Image file name list for comparison. Event is fired when it matches the image

[Goto Table I-133](#)

UPOS Ver1.16 RCSD Specification

				specified here. (Wild card can be specified)
Filters		object	N	
Gender		object	N	Information on gender recognition filter.
TypeList		array	Y	Target Filter TypeList. Valid values are defined by CapTypeList. Recognition target is specified. To disable the filter, null should be assigned in its value.
Score		number	Y	Recognition score. Valid values are from 0 to 100. The range of the score specified here is the recognition target. To disable the filter, -1 should be assigned in its value.
Age		object	N	Information on age recognition.
Min		number	Y	Minimum age. The age below the specified is not a recognition target. To disable the filter -1 should be specified in its value.
Max		number	Y	Maximum age. The age above the specified is not a recognition target. To disable the filter -1 should be specified in its value.
Expression		object	N	Information on facial expression recognition filter.
TypeList		array	Y	Filter target type list. Valid values are defined in CapTypeList. Recognition target type is specified. To disable the filter null should be assigned in its value.
Score		number	Y	Recognition score. Valid values are from 0 to 100. The range of the score specified here is to be recognized. To disable the filter -1 should be assigned in its value.
Gaze		object	N	Information on gaze recognition filter
TypeList		array	Y	Filter target type list. Valid values are defined by CapTypeList. Recognition target is specified. To disable the filter, null should be assigned in its value.
Distance		object	N	Information on distance recognition filter
TypeList		array	Y	Filter target type list. Valid values are defined by CapTypeList. Recognition target is specified. To disable the filter, null should be assigned in its value.

[Goto Table 1-133](#)

UPOS Ver1.16 RCSD Specification
IndividualRecognitionInformation Property Example Format

IndividualRecognitionInformation property of individual recognition device may define various information and here is the example format described by JSON.

■ Basic Items

Key	Value	Value change capability	Explanation
IndividualRecognitionInformation	object	N	Various Individual recognition input data.
[IndividualRecognitionID]	object	N	Store the input data of individual recognition. Key name is ID of individual recognition.
Properties	Array <object>	N	Input data list of target individual recognition. The content of the data is different for each device or function.
[Data01]	-	-	
[Data02]	-	-	

[Goto Table 1-133](#)

UPOS Ver1.16 RCSD Specification

■ **Face Recognition Device Example**

Key	Value	Value change capability	Explanation
IndividualRecognitionInformation	object	N	
Face	object	N	
DataLists	array <object>	N	
FaceID	string	N	ID assigned to the recognized face
FaceImageName	string	N	Recognized face image file name
Gender	object	N	Recognized gender information
Type	string	N	Recognized type
Score	number	N	Confidence score of recognized type.
Age	object	N	Recognized age information
Age	number	N	Recognized age
Expression	object	N	Recognized facial expression information
Type	string	N	Recognized type. One of CapTypeList items is set.
Score	number	N	Confidence score of recognized type.
Gaze	object	N	A gaze list for each recognized face ID.
Type	string	N	Recognized type
Distance	object	N	Recognized distance information
Type	string	N	Recognized type. One of CapTypeList items is set.
Authentication	object	N	Authentication result information
ImageName	string	N	Matched image file name

[Goto Table 1-133](#)

UPOS Ver1.16 RCSD Specification

Properties (UML attributes)

CapIndividualList Property

Syntax	CapIndividualList: <i>string</i> {read-only, access after open}						
Remarks	Recognizable Individual information is indicated by the list separated by a separator ";". Each Individual information consists of the following information and is shown in the following order, separated with a colon (":"). <table><thead><tr><th><u>Parameter</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>IndividualID</td><td>An ID indicated an identifiable Individual</td></tr><tr><td>IndividualName</td><td>A Name of an Individual.</td></tr></tbody></table> This property is initialized by the open method.	<u>Parameter</u>	<u>Meaning</u>	IndividualID	An ID indicated an identifiable Individual	IndividualName	A Name of an Individual.
<u>Parameter</u>	<u>Meaning</u>						
IndividualID	An ID indicated an identifiable Individual						
IndividualName	A Name of an Individual.						
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.						
See Also	IndividualIDs Property on page XX-11 Goto Table 1-34						

IndividualIDs Property

Syntax	IndividualIDs: <i>string</i> {read-only, access after open}
Remarks	Set the IndividualIDs recognizable Individual recognition device. IndividualIDs values are indicated by separated with a colon (":"). Holds an IndividualID recognized by Individual recognition and indicated by separated with a colon (":"). Its value is set prior to a DataEvent being delivered to the application.
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See Also	CapIndividualList Property Goto Table 1-135

IndividualRecognitionFilter Property

Syntax	IndividualRecognitionFilter: <i>string</i> {read-write, access after open}
Remarks	Holds data indicating the following: Individual Recognition Function Information: <ul style="list-style-type: none">Supporting for the various functions (Refer to the Individual Recognition Filter Example Format written by JSON and supported function examples supported function are defined by the device.).Various Valid / Invalid State functions.Various handled function types. Types handled by various functions. (e.g., "male" "female" in gender recognition, etc.).Various filter function settings. Filter setting of various functions. All Individual Recognition function data information is defined by the device. By referring to these contents, the application can determine the supporting scope ete . Thereby, the application can control each function by changing the valid / invalid state and / or the various filter function settings. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20. Goto Table 1-136

UPOS Ver1.16 RCSD Specification
IndividualRecognitionInformation Property

Syntax **IndividualRecognitionInformation: *string* {read-only, access after open}**

Remarks Holds data indicating the following. Individual recognition input data. All Individual recognition input data is defined by the device.

Errors A UposException may be thrown when this property is accessed.
For further information, see “**Errors**” on page Intro-20.

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

DataEvent

<<event>> upos::events::DataEvent

Status : *int32*{read-only}

Description Notifies the application when data from the Individual Recognition device is available to be read.

Attributes This event contains the following attributes:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>Status</i>	<i>int32</i>	<i>Set to 0.</i>

Remarks Before this event is delivered, the individual recognition information is enqueued into the area that is indicated by the **readValue** method. Since the stored individual recognition information might be managed by the associated “**Hard Totals**” device service, therefore, the application might also support the “**Hard Totals**” service.

See Also **VideoCaptureMode** Property, “**Events**” on page Intro-19, **directIO** method.

[Goto Table 1-137](#)

DirectIOEvent

<<event>> upos::events::DirectIOEvent

EventNumber : *int32* {read-only}

Data : *int32* {read-write}

Obj : *object* {read-write}

Description Provides Service information directly to the application. This event provides a means for a vendor-specific Individual Recognition Service to provide events to the application that are not otherwise supported by the device control.

Attributes This event contains the following attributes:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>EventNumber</i>	<i>int32</i>	Event number whose specific values are assigned by the Service.
<i>Data</i>	<i>int32</i>	Additional numeric data. Specific values vary by the <i>EventNumber</i> and the Service. This attribute is settable.
<i>Obj</i>	<i>object</i>	Additional data whose usage varies by the <i>EventNumber</i> and the Service. This attribute is settable.

Remarks This event is to be used only for those types of vendor specific functions that are not otherwise described.

Use of this event may restrict the application program programform being used with other vendor’s devices which may not have any knowledge of the Service’s need for this event.

See Also “**Events**” on page Intro-19, **directIO** method.

[Goto Table 1-137](#)

UPOS Ver1.16 RCSD Specification
ErrorEvent

<<event>> upos::events:: ErrorEvent

ErrorCode : int32 {read-only}
ErrorCodeExtended : int32 {read-only}
ErrorLocus : int32 {read-only}
ErrorResponse : int32{read-write}

Description Notifies the application that an Individual Recognition Device error has been detected and suitable response by the application is necessary to process the error condition.

Attributes This event contains the following attributes:

Attributes	Type	Description
<i>ErrorCode</i>	int32	Error code causing the error event. See a list of Error Codes on page 20.
<i>ErrorCodeExtended</i>	int32	Extended Error code causing the error event. If <i>ErrorCode</i> is E_EXTENDED, then see values below. Otherwise, it may contain a Service-specific value.
<i>ErrorLocus</i>	int32	Location of the error. If EL_OUTPUT is specified. An error occurred during asynchronous action.
<i>ErrorResponse</i>	int32	<i>Error Response, whose default value may be overridden by the application. (i.e., this attribute is settable). See ErrorResponse below for values.</i>

The *ErrorLocus* attribute has one of the following values:

Value	Meaning
EL_OUTPUT	Error occurred while processing asynchronous output.
EL_INPUT	Error occurred while gathering or processing event-driven input. No previously buffered input data is available.
EL_INPUT_DATA	Error occurred while gathering or processing event-driven input, and some previously buffered data is available.

The application's error event handler can set the *ErrorResponse* attribute to one of the following values:

Value	Meaning
ER_RETRY	Retry sending the data. The error state is exited. May be valid for some input devices when the locus is EL_INPUT, in which case the input is re-tried, and the error state is exited. Typically, valid for asynchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited. This is the default response when the locus is EL_OUTPUT.
ER_CLEAR	Valid for all loci: EL_INPUT, EL_INPUT_DATA, and EL_OUTPUT. Clear all buffered input or output data (including all asynchronous output). The error state is exited. This is the default response when the locus is EL_INPUT.

UPOS Ver1.16 RCSD Specification

ER_CONTINUEINPUT

Only valid when the locus is EL_INPUT_DATA.
Acknowledges that a data error has occurred and directs the Device to continue input processing. The Device remains in the error state and will deliver additional **DataEvents** as directed by the **DataEventEnabled** property. When all input has been delivered and **DataEventEnabled** is again set to true, then another **ErrorEvent** is delivered with locus EL_INPUT.
This is the default response when the locus is EL_INPUT_DATA.

Remarks This event is enqueued when an error is detected and the Device's **State** transitions into the error state. Input error events are not delivered until **DataEventEnabled** is true, so that proper application sequencing occurs.

Unlike a **DataEvent**, the Device does not disable further **DataEvents** or input **ErrorEvents**; it leaves the **DataEventEnabled** property value at true. Note that the application may set **DataEventEnabled** to false within its event handler if subsequent input events need to be disabled for a period of time.

See Also "Device Input Model" on page Intro-22, "Error Handling" on page Intro-23, "Device Output Models" on page Intro-25.

[Goto Table 1-139](#)

StatusUpdateEvent

<<event>> upos::events:: StatusUpdateEvent
Status : int32 {read-only}

Description *Notifies the application that there is a change in the power status or a status of the Individual Recognition device.*

Attributes This event contains the following attribute:

Attribute	Type	Description
-----------	------	-------------

Status	int32	Indicates a change in the power status of the unit.
--------	-------	---

Note that Release 1.3 added Power State Reporting with additional Power reporting StatusUpdateEvent values.

The Update Firmware capability added additional Status values

For communicating the status/progress of an asynchronous update firmware process. See "StatusUpdateEvent" description on page 1-34.

Remarks Enqueued when the Individual Recognition Device detects a power state change or a status change.

See Also "Events" on page Intro-19

[Goto Table 1-140](#)

Sound Recorder

This Chapter defines the Sound Recorder device category.

Summary

Properties(UML attributes)

<i>Common</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
AutoDisable:	<i>boolean</i>	{read-write}	1.16	open
CapCompareFirmwareVersion:	<i>boolean</i>	{read-only}	1.16	open
CapPowerReporting:	<i>int32</i>	{read-only}	1.16	open
CapStatisticsReporting:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateFirmware:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateStatistics:	<i>boolean</i>	{read-only}	1.16	open
CheckHealthText:	<i>string</i>	{read-only}	1.16	open
Claimed:	<i>boolean</i>	{read-only}	1.16	open
DataCount:	<i>int32</i>	{read-only}	1.16	open
DataEventEnabled:	<i>boolean</i>	{read-write}	1.16	open
DeviceEnabled:	<i>boolean</i>	{read-write}	1.16	open & claim
FreezeEvents:	<i>boolean</i>	{read-write}	1.16	open
OutputID:	<i>int32</i>	{read-only}	1.16	Not supported
PowerNotify:	<i>int32</i>	{read-write}	1.16	open
PowerState:	<i>int32</i>	{read-only}	1.16	open
State:	<i>int32</i>	{read-only}	1.16	open
DeviceControlDescription:	<i>string</i>	{read-only}	1.16	--
DeviceControlVersion:	<i>int32</i>	{read-only}	1.16	--
DeviceServiceDescription:	<i>string</i>	{read-only}	1.16	open
DeviceServiceVersion:	<i>int32</i>	{read-only}	1.16	open
PhysicalDeviceDescription:	<i>string</i>	{read-only}	1.16	open
PhysicalDeviceName:	<i>string</i>	{read-only}	1.16	open

UPOS Ver1.16 RCSD Specification

Properties (Continued)

<i>Specific</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
CapAssociatedHardTotalsDevice:	string	{read-only}	1.16	open
CapChannel:	boolean	{read-only}	1.16	open
CapChannelList:	string	{read-only}	1.16	open
CapRecordingLevel:	boolean	{read-only}	1.16	open
CapSamplingRate:	boolean	{read-only}	1.16	open
CapSamplingRateList:	string	{read-only}	1.16	open
CapSoundType:	boolean	{read-only}	1.16	open
CapSoundTypeList:	string	{read-only}	1.16	open
CapStorage	int32	{read-only}	1.16	open
Channel:	string	{read-write}	1.16	open, claim & enable
ChannelList:	string	{read-only}	1.16	open
RecordingLevel:	int32	{read-write}	1.16	open, claim & enable
RemainingRecordingTimeInSec:	int32	{read-only}	1.16	open, claim & enable
SamplingRate:	string	{read-write}	1.16	open, claim & enable
SamplingRateList:	string	{read-only}	1.16	open
SoundData:	binary	{read-only}	1.16	open
SoundType:	string	{read-write}	1.16	open, claim & enable
SoundTypeList:	string	{read-only}	1.16	open
Storage	int32	{read-write}	1.16	open, claim & enable

Methods(UML operations)

Common

<i>Name</i>	<i>Version</i>
open (logicalDeviceName: string): void {raises-exception}	1.16
close (): void {raises-exception, use after open}	1.16
claim (timeout: int32): void {raises-exception, use after open}	1.16
release (): void {raises-exception, use after open, claim}	1.16
checkHealth (level: int32): void {raises-exception, use after open, enable}	1.16
clearInput (): void {}	1.16 Not supported

[Goto Table 1-141](#) [Goto Table 1-142](#)
[Goto Table 1-143](#) [Goto Table 1-144](#)

UPOS Ver1.16 RCSD Specification

Methods (UML operations)(continued)

Common

<i>Name</i>	<i>Version</i>
clearInputProperties (): void { }	1.16 Not supported
clearOutput (): void { }	Not supported
compareFirmwareVersion (firmwareFileName: string, out result: int32): void {raises-exception, use after open, enable}	1.16
directIO (command: int32, inout data: int32, inout obj: object): void {raises-exception, use after open}	1.16
resetStatistics (statisticsBuffer: string): void {raises-exception, use after open, enable}	1.16
retrieveStatistics (inout statisticsBuffer: string): void {raises-exception, use after open, enable}	1.16
updateFirmware (firmwareFileName: string): void {raises-exception, use after open, enable}	1.16
updateStatistics (statisticsBuffer: string): void {raises-exception, use after open, enable}	1.16

Specific

<i>Name</i>	<i>Version</i>
startRecording (FileName: string, OverWrite: boolean, RecordingTime:int32): void {raises-exception, use after open, claim, enable}	1.16
stopRecording (): Void {raises-exception, use after open, claim, enable}	1.16

[Goto Table 1-145](#)

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

<i>Name</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>
upos::events::DataEvent			1.16
Status:	<i>int32</i>	{read-only}	
upos::events::DirectIOEvent			1.16
EventNumber:	<i>int32</i>	{read-only}	
Data:	<i>int32</i>	{read-write}	
Obj:	<i>object</i>	{read-write}	
upos::events::ErrorEvent			1.16
ErrorCode:	<i>int32</i>	{read-only}	
ErrorCodeExtended:	<i>int32</i>	{read-only}	
ErrorLocus:	<i>int32</i>	{read-only}	
*pErrorResponse:	<i>int32</i>	{read-write}	
upos::events::OutputCompleteEvent		<i>Not supported</i>	1.16
upos::events::StatusUpdateEvent			1.16
Status:	<i>int32</i>	{read-only}	
upos::events::TransitionEvent		<i>Not supported</i>	1.16

[Goto Table 1-146](#)

[Goto Table 1-147](#)

UPOS Ver1.16 RCSD Specification
General Information

The Sound Recorder programmatic name is "Sound Recorder".

Capabilities

The Sound Recorder has the following capability:

~~Save the recorded sound to a file:~~

Record the real-time audio to a file, deliver the recorded sound data to the property that application may read and / or retrieve, and save the recorded sound data file to device memory and / or other storage devices.

[Goto Table 1-148](#)

Sound Recorder Class Diagram

The following diagram shows the relationships between the Sound Recorder classes.

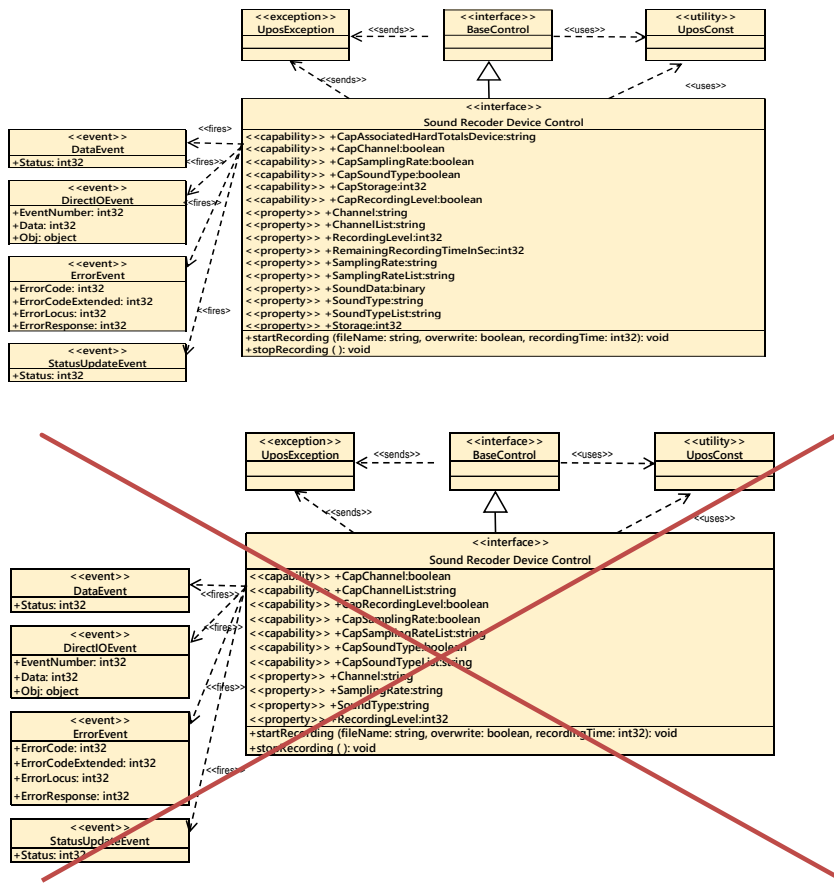


Fig. Chap. 41-1 Sound Recorder Class Diagram

[Goto Table2-5](#)

UPOS Ver1.16 RCSD Specification Model

Sound Recorder Control follows a general “Device Input Model” ~~for event driven input~~ in a broad sense. One point of difference is that the Sound Recorder device required the execution of methods to start and stop the sound recording process and creates a sound data file in real time, deliver the data to the property and save the file in device and / or associated storage device.

The Sound Recorder Model defines the following behavior: Sound Recorder device controls the Sound Recorder device to set the input (recording) conditions, specifies the start / end of input data acquisition by the method. And makes the sound data file in real time from the acquired audio and delivers the data to the appropriate property. At the same time, saves the recorded data file in device and /or associated storage devices.

“Sound Recorder” device control starts recording with the **startRecording** method.

Prior to execute the **startRecording** method each value setting of **Channel** property, **SamplingRate** property, and **RecordingLevel** property are required, if each of **CapChannel** property **CapSamplingRate** property is true. And also need to set the **DataEventEnabled** property to true. At the same time, the recording format setting starts with the **SoundType** property value, if **CapSoundType** property is true.

The recording ends after the specified time has elapsed or when **stopRecording** method is called or when **clearInput** method is called. The generated sound data file will be recorded for either the host file or the Hard Totals device or both, after the end of recording. And generated sound data will be delivered to the **SoundData** property. Just after the delivery of sound data to the property, when the **DataEventEnabled** property is true, the **DataEvent** is enqueued and delivered to the application.

~~•“The control will generate a **DataEvent** when the recording started by the **startRecording** method ends when the specified time elapses and the recording to the specified file is completed.”~~

~~•“When an application calls the **stopRecording** method to end recording, **DataEvent** will not occur.”~~

If the **AutoDisable** property is true, the device will automatically disable itself after the **DataEvent** is enqueued.

The remaining recording time in seconds can be obtained from the property **RemainingRecordingTimeInSec**

StatusUpdateEvent with status **SERC_SUE_START_SOUND_RECORDING** is evoked when **startRecording** method is executed to notify the application that recording state with has started.

When the sound recording is finished, if the specified time of **startRecording** method has elapsed or **stopRecording** method has been called, a **StatusUpdateEvent** with

[Goto Table 1-149](#)

UPOS Ver1.16 RCSD Specification

status `SERC_SUE_STOP_SOUND_RECORDING` is evoked to notify the application that recording has been stopped.

~~* An **ErrorEvent** (or events) is enqueued if an error occurs while gathering or processing input, and is delivered to the application when **DataEventEnabled** is true and other event delivery requirements are met.~~

~~* The **DataCount** property may be read to obtain the total number of enqueued **DataEvents**.~~

~~* All enqueued input may be deleted by calling **clearInput**. See the **clearInput** method description for more details.~~

~~* All data properties that are populated as a result of firing a **DataEvent** or **ErrorEvent** can be set back to their default values by calling the **clearInputProperties** method.~~

~~* Since audio files are recorded in the area managed by the "hard total" service, the application must also support "hard total" services.~~

An enqueued **DataEvent** can be delivered to the application when the **DataEventEnabled** property is true and other event delivery requirements are met. Just before enqueueing this event, the device provides the recorded data to the **SoundData** property, and disables further data events by setting the **DataEventEnabled** property to false. This causes subsequent input data to be buffered by the device while the application processes the current input and associated properties. When the application has finished processing the current input and is ready for more data, it re-enables events by setting **DataEventEnabled** to true.

If **ErrorEvent** response is `ER_CONTINUEINPUT`, the process of input processing continues. However, as long as the cause of the error is not resolved, the **ErrorEvent** will occur again immediately.

If **ErrorEvent** is `ER_CLEAR`, the input processing process is terminated, and the record is discarded.

If the time specified by the **startRecording** method is `FOREVER (-1)`, execution will continue until the **stopRecording** method is called in the application. When **stopRecording** is called, the previous recording data is recorded to the host file, the Hard Totals device, or both, with the specified file name, and the sound data will be delivered to the **SoundData** property. When **DataEventEnabled** property is true, the **DataEvent** is enqueued and delivered to the application.

Only one call to **startRecording** method can be in progress at a time. An attempt to nest sound recorder operations will result in an **UposException** being thrown.

If Error occurs during the execution of the **startRecording** method application should call the **stopRecording** method to terminate the recording process or cancel the recording process by calling the **clearInput** method before ending the **ErrorEvent** processing. After this when the **stopRecording** method is called, the recording data until just before the **ErrorEvent** occurs is recorded to the host file, the Hard Totals device, or both. When **DataEventEnabled** property is true, the **DataEvent** is enqueued and delivered to the application.

If there is no Error during the execution of **startRecording** method can terminate the recording process and can stop the recording at any time. When the **stopRecording** method is called, the recording data until just before the method call is recorded to the host file, the Hard Totals or both. When **DataEventEnabled** property is true, the **DataEvent** is enqueued and delivered to the application.

All input data enqueued by the device may be deleted by calling the **clearInput** method. All data properties that are populated as a result of a **DataEvent** or **ErrorEvent** can be set back to their default values by calling the **clearInputProperties**

UPOS Ver1.16 RCSD Specification

method.

The device may have the ability to write encoded sound data files to either the Hard Totals devices or the host file system, or both, and the **CapStorage** property will show the device's data storage location capability.

If device supports either or both Hard Totals devices and the host file system, the application should set the **Storage** property accordingly to tell where to write the encoded sound data file.

If device needs to be able to write the encoded sound data to an associated Hard Totals device, the **CapAssociatedHardTotalsDevice** property holds the open name of the associated Hard Totals device.

[Goto Table 1-149](#)

Device Sharing

The Sound Recorder is an exclusive-use device, as follows:

- The application must claim the device before enabling it.
- The application must claim and enable the device before accessing some properties or calling methods that update the device.
- See the "Summary" table for precise usage prerequisites.
- The image display mode of the graphics device control is as follows.

UPOS Ver1.16 RCSD Specification

Properties(UML attributes)

CapAssociatedHardTotalsDevice Property

Syntax	CapAssociatedHardTotalsDevice : <i>string</i> {read-write, access after open}
Remarks	Holds the open name of the associated Hard Totals device if the device is able to write to such devices which is the case if CapStorage is either SREC_CST_ALL or SREC_CST_HARDTOTALS_ONLY. If CapStorage is SREC_CST_HOST_ONLY this property value must be the empty string.
Errors	UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See Also	CapStorage Property Goto Table 1-150

CapChannel Property

Syntax	CapChannel : <i>boolean</i> {read-only, access after open}
Remarks	If true, the application can change the channel. If false, the application cannot change the channel. This property is initialized by the open method.
Errors	UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See Also	Channel Property

~~CapChannelList Property~~

Syntax	CapChannelList : <i>string</i> {read-only, access after open}
Remarks	Contains the comma-delimited list of channel that is supported by the device. For example, if the device only supports 1ch and 2ch and 4ch, then this property should be set to "1,2,4". This property is initialized by the open method.
Errors	UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See Also	Channel Property.

[Goto Table 1-151](#)

CapSamplingRate Property

Syntax	CapSamplingRate : <i>boolean</i> {read-only, access after open}
Remarks	If true, the application can change the sampling rate. If false, the application cannot change the sampling rate. This property is initialized by the open method.
Errors	UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.
See Also	SamplingRate Property.

UPOS Ver1.16 RCSD Specification

CapSamplingRateList Property

Syntax ~~CapSamplingRateList + string {read-only, access after open}~~

Remarks ~~Contains the comma-delimited list of sampling rate that are supported by the device.~~

~~For example, if the device only supports 44.1KHz and 48KHz and 96KHz, then this property should be set to "44100,48000,96000".~~

~~This property is initialized by the open method.~~

Errors ~~UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.~~

See Also ~~SamplingRate Property.~~

[Goto Table 1-152](#)

CapSoundType Property

Syntax CapSoundType: *boolean* {read-only, access after open}

Remarks If true, the application can change the sound file type.
If false, the application cannot change the sound file type.
This property is initialized by the **open** method.

Errors UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.

See Also SoundType Property.

CapSoundTypeList Property

Syntax ~~CapSoundTypeList + string {read-only, access after open}~~

Remarks ~~Contains the comma-delimited list of sound file type that is supported by the device.~~

~~For example, if the device only supports WAV and OGG, then this property should be set to "WAV,OGG".~~

~~This property is initialized by the open method.~~

Errors ~~UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.~~

See Also ~~SoundType Property.~~

[Goto Table 1-153](#)

UPOS Ver1.16 RCSD Specification

CapStorage Property

Syntax **CapStorage: *int32* {read-only, access after open}**

Remarks This is an enumeration and announces where the device is able to write the recorded sound data file to.
It holds one of the following values.

<u>Value</u>	<u>Meaning</u>
SREC_CST_HARDTOTALS_ONLY	Only an associate Hard Totals device is supported.
SREC_CST_HOST_ONLY	Only the host's file system is supported.
SREC_CST_ALL	Both, the associated Hard Totals device and the host's file system is supported.

This property is initialized by the **open** method.

If a Hard Totals device is supported the **Storage** the property value should be SREC_CST_HARDTOTALS_ONLY or SREC_CST_ALL, and the property **CapAssociatedHardTotalsDevice** holds the open name of the associated **Hard Totals** device.

Errors UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.

See Also **Storage** Property, **CapAssociatedHardTotalsDevice** Property

[Goto Table 1-154](#)

CapRecordingLevel Property

Syntax **CapRecordingLevel: *boolean* {read-only, access after open}**

Remarks If true, the application can change the recording level.
If false, the application cannot change the recording level.
This property is initialized by the **open** method.

Errors UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.

See Also **CapRecordingLevel** Property.

Channel Property

Syntax **Channel : *string* {read-write, access after open-claim-enable}**

Remarks Holds the channel during recording.
Valid values are one of the values listed in the **ChannelList** property.
This property is initialized by the **open** method.

Errors UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.
E_BUSY	Property could not be set because it is recording.

See Also **CapChannel** Property, **ChannelList** Property

[Goto Table 1-155](#)

UPOS Ver1.16 RCSD Specification

ChannelList Property

- Syntax** **ChannelList** : *string* {read only, access after open}
- Remarks** Contains the comma-delimited list of channels that is supported by the device.

For example, if the device only supports channel1 and channel2 and channel4, then this property should be set to "1,2,4".
This property is initialized by the **open** method.
- Errors** UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.
- See Also** **Channel Property**. [Goto Table 1-156](#)

RecordingLevel Property

- Syntax** **RecordingLevel** : *int32* {read-write, access after open- claim-enable}
- Remarks** Holds the recording level during recording.
Legal values range from zero through 100.
This property is initialized by the **open** method.
- Errors** UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.
- | Value | Meaning |
|-----------|---------------------------------|
| E_ILLEGAL | An invalid value was specified. |
- See Also** **CapRecordingLevel Property**

RemainingRecordingTimeInSec Property

- Syntax** **RemainingRecordingTimeInSec**:
 int32 {read-only, access after open-claim-enable}
- Remarks** This property holds the remaining recording time in seconds if a recording is ongoing. If no recording is ongoing its value is 0. When a call to method **startRecording** returns, this property initially holds the time passed as argument *recordingTime* to that call. If this argument value is FOREVER, this property also holds this value unchanged until **stopRecording** has been called.

This property is initialized during device **setDeviceEnabled** method to 0.
- Errors** UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.
- See Also** **startRecording Method, stopRecording Method** [Goto Table 1-157](#)

SamplingRate Property

- Syntax** **SamplingRate** : *string*{read-write, access after open-claim-enable}
- Remarks** Holds the sampling rate during recording.
Valid values are one of the values listed in the **SamplingRateList** property.
This property is initialized by the **open** method.
- Errors** UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.
- | Value | Meaning |
|-----------|--|
| E_ILLEGAL | An invalid value was specified. |
| E_BUSY | Property could not be set because it is recording. |
- See Also** **CapSamplingRate Property, SamplingRateList Property**

[Goto Table 1-158](#)

UPOS Ver1.16 RCSD Specification

SamplingRateList Property

- Syntax** `SamplingRateList : string {read only, access after open}`
- Remarks** Contains the comma-delimited list of sampling rate that are supported by the device.
For example, if the device only supports 44.1kHz and 48kHz and 96kHz, then this property should be set to "44100,48000,96000".
This property is initialized by the **open** method.
- Errors** UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.
- See Also** [SamplingRate](#) Property. [Goto Table 1-159](#)

SoundData Property

- Syntax** `SoundData: binary { read-only, access after open }`
- Remarks** This property is used to store the sound data after the recording time elapse of startRecording method or stopRecording method is called. If no recorded sound data was available, the **SoundData** property will be set to zero length (or empty). Its value is set prior to a **DataEvent** to be enqueued. This property is initialized to zero length by the **open** method.
- Errors** A UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-21.
- See Also** [startRecording](#) Method, [stopRecording](#) Method, [DataEvent](#). [Goto Table 1-160](#)

SoundType Property

- Syntax** `SoundType : string {read-write, access after open, claim}`
- Remarks** Holds the audio file format to be recorded.
Valid values are one of the values listed in the **CapSoundTypeList** property.
This property is initialized by the open method.
- Errors** UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.
- | <u>Value</u> | <u>Meaning</u> |
|----------------------|---|
| E_ILLEGAL | An invalid value was specified. |
| E_BUSY | Property could not be set because it is recording. |
- See Also** [CapSoundType](#) Property, [CapSoundTypeList](#) Property. [Goto Table 1-161](#)

SoundTypeList Property

- Syntax** `SoundTypeList : string{read only, access after open}`
- Remarks** Contains the comma-delimited list of sound file type that is supported by the device.
For example, if the device only supports WAV and OGG, then this property should be set to "WAV,OGG".
This property is initialized by the **open** method.
- Errors** UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.
- See Also** [SoundType](#) Property. [Goto Table 1-162](#)

UPOS Ver1.16 RCSD Specification

Storage Property

Syntax Storage: *int32* {read-write, access after open-claim-enable}

Remarks This is an enumeration and defines where the device writes the recorded sound data file to. Should be set before a call to **startRecording**. It holds one of the following values.

<u>Value</u>	<u>Meaning</u>
SREC_ST_HARDTOTALS	The encoded data file is written to the associated Hard Totals device. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device.
SREC_ST_HOST	The encoded data file is written to the host's file system.
SREC_ST_HOST_HARDTOTALS	The encoded data file is written to the associated Hard Totals device and host's file system. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device.

This property is initialized by the **open** method according to the value hold by **CapStorage**. If **CapStorage** has the value SREC_CST_ALL, it is initialized to SREC_ST_HOST_HARDTOTALS.

Errors UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified, or recording is ongoing.

See Also **CapStorage** Property, **CapAssociatedHardTotalsDevice** Property

[Goto Table 1-163](#)

Methods(UML operations)

startRecording Method

Syntax **startRecording** (**fileName**: *string*, **overWrite**: *boolean*, **recordingTime** : *int32*):
void{raises-exception, use after open-claim-enable}

Parameter	Description
<i>fileName</i>	Specify the file name of the sound to be recorded.
<i>overWrite</i>	Specify the behavior when the same name file exists. If it is true it will be overwritten and if false it will raise the UPOSException return-an-error .
<i>recordingTime</i>	Specify the time for recording in seconds. If OPOS_FOREVER (-1) is specified, recording will continue until you call the stopRecording method is called.

Remarks ~~Recording starts with the settings of the **Channel** property, **SamplingRate** property, and **RecordingLevel** property, and recording starts in the format set by **SoundType**.~~

Sound recording starts with the settings of the **Channel** property, **SamplingRate** property, and **RecordingLevel** property and need to set **DataEventEnabled** property to true. At the same time, recording format setting starts with the **SoundType** property. When this method is called, if specified recording time is elapsed, recording process will be ended and recorded sound data is provided at the **SoundData** property that the application may read it and / or process the stored sound data file given as *filename* argument. When the **DataEventEnabled** property is true, the **DataEvent** is enqueued and delivered to the application. **StatusUpdateEvent** with state **SREC_SUE_START_SOUND_RECORDING** is evoked when **startRecording** method is executed to notify the application, the recording has started. When the sound recording is finished, if the specified time of **startRecording** method has elapsed or **stopRecording** method has been called, the value of **StatusUpDateEvent** with state **SREC_SUE_STOP_SOUND_RECORDING** is evoked to notify the application, the recording has stopped

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20

Some possible values of the exception's *ErrorCode* property are:

Value	Meaning
E_ILLEGAL	FileName is too long or contains characters that cannot be used, or 0 is specified for RecordingTime.
E_EXISTS	FileName already exists. (When OverWrite is FALSE)
E_BUSY	It cannot be executed as it is recording.

See Also **Channel** Property, **SamplingRate** Property, **SoundData** Property, **SoundType** Property, **RecordingLevel** Property, **stopRecording** Method, **StatusUpdateEvent** Event

[Goto Table 1-164](#)

UPOS Ver1.16 RCSD Specification

stopRecording Method

Syntax **stopRecording ():**
 void {raises-exception, use after open-claim-enable}

Remarks ~~Finish the recording and complete the recording of the audio file.~~
When this method is called the sound recording process that started by **startRecording** method is ended and the recording is finished. This method is processed synchronously. After recording and decoding process has been finished, the recorded sound data will be provided at the **SoundData** property prior to the Data Event is enqueued, when **DataEventEnabled** property is true. When **stopRecording** method is called, a **StatusUpdateEvent** with status **SREC_SUE_STOP_SOUND_RECORDING** is evoked to notify the application, the recording has stopped.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20 Some possible values of the exception’s *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	It is not recorded.

See Also **StartRecording** Property, **SoundData** Property, **StatusUpdateEvent**. Event

[Goto Table 1-165](#)

UPOS Ver1.16 RCSD Specification

Events(UML interfaces)

DataEvent

<<event>> upos::events::DataEvent

Status :int32{read-only}

Description Notifies the application when data from the Sound Recorder device is available to be read.

Attributes This event contains the following attributes:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>Status</i>	<i>int32</i>	<i>Set to 0.</i>

Remarks Before this event is delivered, the Sound Recorder information is enqueued into the area that is indicated by the startRecording method. Since the stored sound recorder device information might be managed by the associated "Hard Totals" device service, therefore, the application might also support the "Hard Totals" service.

See Also **Channel** Property, **SamplingRate** Property, **SoundType** property, **RecordingLevel** Property, **stopRecording** Method, **startRecording** Method

[Goto Table 1-166](#)

DirectIOEvent

<<event>> upos::events::DirectIOEvent

EventNumber : int32 {read-only}
Data : int32 {read-write}
Obj : object {read-write}

Description Provides Service information directly to the application. This event provides a means for a vendor-specific Individual Recognition Service to provide events to the application that are not otherwise supported by the device control.

Attributes This event contains the following attributes:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>EventNumber</i>	<i>int32</i>	Event number whose specific values are assigned by the Service.
<i>Data</i>	<i>int32</i>	Additional numeric data. Specific values vary by the <i>EventNumber</i> and the Service. This attribute is settable.
<i>Obj</i>	<i>object</i>	Additional data whose usage varies by the <i>EventNumber</i> and the Service. This attribute is settable.

Remarks This event is to be used only for those types of vendor specific functions that are not otherwise described.

Use of this event may restrict the application program programform being used with other vendor's devices which may not have any knowledge of the Service's need for this event.

See Also "Events" on page Intro-19, **directIO** method.

[Goto Table 1-167](#)

UPOS Ver1.16 RCSD Specification
ErrorEvent

```
<<event>> upos::events:: ErrorEvent
ErrorCode : int32{read-write}
ErrorCodeExtended : int32{read-write}
ErrorLocus : int32{read-write}
*pErrorResponse : int32{read-write}
```

Description Notifies the application that a Sound Recorder Device error has been detected and suitable response by the application is necessary to process the error condition.

Attributes This event contains following attributes.

Attributes	Type	Description
<i>Error Code</i>	<i>int32</i>	Error Code causing the error event. See the list of Error Code.
<i>ErrorCodeExtended</i>	<i>int32</i>	Error Code causing the error event. These values are device category specific.
<i>ErrorLocus</i>	<i>int32</i>	Location of the error. See values below.
<i>pErrorResponse</i>	<i>int32</i>	Pointer to the error event response. See <i>ErrorResponse</i> values below.

The ErrorLocus attribute has one of the following values:

Value	Meaning
EL_INPUT	Error occurred while gathering or Processing event-driven input. No previously buffered input data is available.
EL_INPUT_DATA	Error occurred while gathering or processing event-driven input, and some previously buffered data is available.

If ResultCode is E_EXTENDED, ResultCodeExtended is set to one of the following values.

Value	Meaning
ESRECEPTOF_NOROOM	There is not enough space to store the data file.

The application's error event handler can set the ErrorResponse attribute to one of the following values:

Value	Meaning
ER_CLEAR	I will try its asynchronous output again. The error condition is exited.
ER_CONTINUEINPUT	Only valid when the locus is EL_INPUT_DATA. Acknowledges that a data error has occurred and directs the Device to continue input processing. The Device remains in the error state and will deliver additional DataEvents as directed by the DataEventEnabled property. When all input has been delivered and DataEventEnabled is again set to true, then another ErrorEvent is delivered with locus EL_INPUT. This is the default response when the locus is EL_INPUT_DATA.

Remarks It notifies you when an error is detected during recording. Input error events are not delivered until **DataEventEnabled** is true, so that proper application sequencing occurs.

See Also [Status, Error code, State model](#) [Goto Table 1-168](#)

UPOS Ver1.16 RCSD Specification
StatusUpdateEvent

<<event>>	upos::events:: StatusUpdateEvent Status : <i>int32</i> {read-only}												
Description	<i>Notifies the application that there is a change in the power status or a status of the Sound Recorder device.</i>												
Attributes	This event contains the following attribute: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Attributes</th> <th style="text-align: left;">Type</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td><i>Status</i></td> <td><i>int32</i></td> <td>Indicates a change in the power status or a status of the unit.</td> </tr> </tbody> </table> <p><i>Note that Release 1.3 added Power State Reporting with additional Power reporting StatusUpdateEvent values.</i></p> <p>The Update Firmware capability added additional <i>Status</i> values for communicating the status/progress of an asynchronous update firmware process. See “StatusUpdateEvent” description on page 1-34.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Value</th> <th style="text-align: left;">Meaning</th> </tr> </thead> <tbody> <tr> <td>SREC_SUE_START_SOUND_RECORDING</td> <td>It will be notified when sound recording starts.</td> </tr> <tr> <td>SREC_SUE_STOP_SOUND_RECORDING</td> <td>It will be notified when sound recording stops.</td> </tr> </tbody> </table>	Attributes	Type	Description	<i>Status</i>	<i>int32</i>	Indicates a change in the power status or a status of the unit.	Value	Meaning	SREC_SUE_START_SOUND_RECORDING	It will be notified when sound recording starts.	SREC_SUE_STOP_SOUND_RECORDING	It will be notified when sound recording stops.
Attributes	Type	Description											
<i>Status</i>	<i>int32</i>	Indicates a change in the power status or a status of the unit.											
Value	Meaning												
SREC_SUE_START_SOUND_RECORDING	It will be notified when sound recording starts.												
SREC_SUE_STOP_SOUND_RECORDING	It will be notified when sound recording stops.												
Remarks	Enqueued when the Sound Recorder Device detects a power state change or a status change.												
See Also	“Events” on page Intro-19. Goto Table 1-169												

Voice Recognition

This Chapter defines the Voice Recognition device category.

Summary

Properties (UML attributes)

<i>Common</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
AutoDisable:	<i>boolean</i>	{read-write}	1.16	open
CapCompareFirmwareVersion:	<i>boolean</i>	{read-only}	1.16	open
CapPowerReporting:	<i>int32</i>	{read-only}	1.16	open
CapStatisticsReporting:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateFirmware:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateStatistics:	<i>boolean</i>	{read-only}	1.16	open
CheckHealthText:	<i>string</i>	{read-only}	1.16	open
Claimed:	<i>boolean</i>	{read-only}	1.16	open
DataCount:	<i>int32</i>	{read-only}	1.16	open
DataEventEnabled:	<i>boolean</i>	{read-write}	1.16	open
DeviceEnabled:	<i>boolean</i>	{read-write}	1.16	open & claim
FreezeEvents:	<i>boolean</i>	{read-write}	1.16	open
OutputID:	<i>int32</i>	{read-only}	1.16	Not supported
PowerNotify:	<i>int32</i>	{read-write}	1.16	open
PowerState:	<i>int32</i>	{read-only}	1.16	open
State:	<i>int32</i>	{read-only}	1.16	--
DeviceControlDescription:	<i>string</i>	{read-only}	1.16	--
DeviceControlVersion:	<i>int32</i>	{read-only}	1.16	--
DeviceServiceDescription:	<i>string</i>	{read-only}	1.16	open
DeviceServiceVersion:	<i>int32</i>	{read-only}	1.16	open
PhysicalDeviceDescription:	<i>string</i>	{read-only}	1.16	open
PhysicalDeviceName:	<i>string</i>	{read-only}	1.16	open

UPOS Ver1.16 RCSD Specification

Properties (Continued)

<i>Specific</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
CapLanguage:	boolean	{read-only}	1.16	open
HearingDataPattern:	string	{read-only}	1.16	open, claim & enable
HearingDataWord:	string	{read-only}	1.16	open, claim & enable
HearingDataWordList:	string	{read-only}	1.16	open, claim & enable
HearingResult:	int32	{read-only}	1.16	open, claim & enable
HearingStatus:	int32	{read-only}	1.16	open, claim & enable
LanguageList:	string	{read-only}	1.16	open

Methods (UML operations)

Common

<i>Name</i>	<i>Version</i>
open (logicalDeviceName: <i>string</i>): void {raises-exception}	1.16
close (): void {raises-exception, use after open}	1.16
claim (timeout: <i>int32</i>): void {raises-exception, use after open}	1.16
release (): void {raises-exception, use after open, claim}	1.16
checkHealth (level: <i>int32</i>): void {raises-exception, use after open, enable}	1.16
clearInput (): void { }	1.16 Not supported
clearInputProperties (): void { }	1.16 Not supported
clearOutput (): void { }	Not supported
compareFirmwareVersion (firmwareFileName: <i>string</i> , out result: <i>int32</i>): void {raises-exception, use after open, claim, enable}	1.16
directIO (command: <i>int32</i> , inout data: <i>int32</i> , inout obj: <i>object</i>): void {raises-exception, use after open}	1.16
resetStatistics (statisticsBuffer: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16
retrieveStatistics (inout statisticsBuffer: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16

[Goto Table 1-170](#)

[Goto Table 1-171](#)

UPOS Ver1.16 RCSD Specification

Methods (UML operations)(continued)

Common

<i>Name</i>	<i>Version</i>
updateFirmware (firmwareFileName: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16
updateStatistics (statisticsBuffer: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16

Specific

<i>Name</i>	
startHearingFree (language: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16
startHearingSentence (language: <i>string</i> , wordList: <i>string</i> , patternList: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16
startHearingWord (language: <i>string</i> , wordList: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16
startHearingYesNo (language: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16
stopHearing (): void {raises-exception, use after open, claim, enable}	1.16

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

<i>Name</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>
upos::events::DataEvent			1.16
Status:	<i>int32</i>	{read-only}	
upos::events::DirectIOEvent			1.16
EventNumber:	<i>int32</i>	{read-only}	
Data:	<i>int32</i>	{read-write}	
Obj:	<i>object</i>	{read-write}	
upos::events::ErrorEvent			1.16
ErrorCode:	<i>int32</i>	{read-only}	
ErrorCodeExtended:	<i>int32</i>	{read-only}	
ErrorLocus:	<i>int32</i>	{read-only}	
ErrorResponse:	<i>int32</i>	{read-write}	
upos::events::OutputCompleteEvent		<i>Not supported</i>	
upos::events::StatusUpdateEvent			1.16
Status:	<i>int32</i>	{read-only}	
upos::events::TransitionEvent		<i>Not supported</i>	1.16

[Goto Table 1-172](#)

UPOS Ver1.16 RCSD Specification

General Information

The Voice Recognition programmatic name is "Voice Recognition".

Capabilities

The Voice Recognition has the following capability:

- Convert spoken words to strings.

Voice Recognition Class Diagram

The following diagram shows the relationships between the Voice Recognition classes.

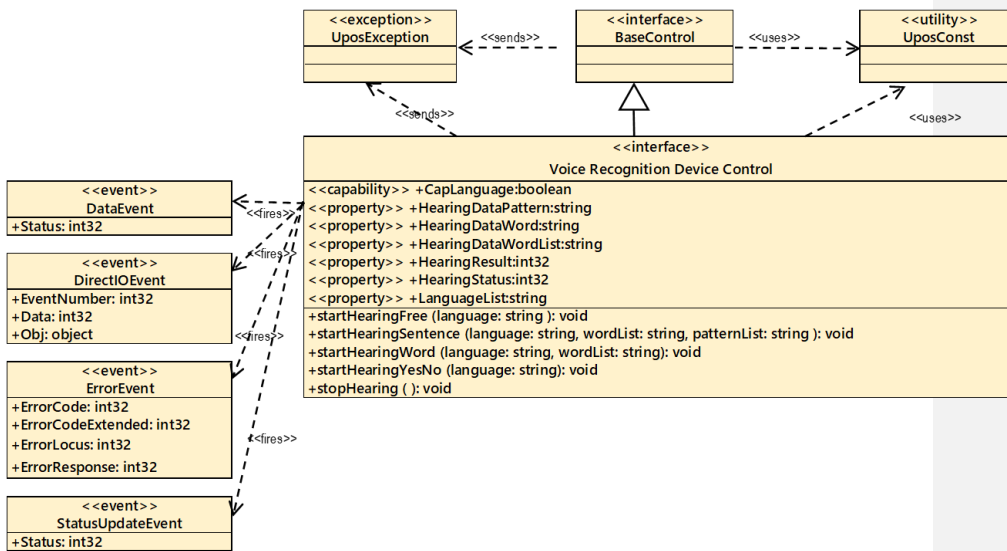


Fig. Chap. 42-1 Voice Recognition Class Diagram

UPOS Ver1.16 RCSD Specification Model

The Voice Recognition follows the general "Device Input Model" for event-driven input:

Device Control starts voice recognition with the **startHearingYesNo** method, **startHearingSentence** method, etc., and generates **DataEvent** when recognizing voice.

If the **AutoDisable** property is true, then the device automatically disables itself when a **DataEvent** is enqueued.

An enqueued **DataEvent** can be delivered to the application when the **DataEventEnabled** property is true and other event delivery requirements are met. Just before delivering this event, data is copied into corresponding properties, and further data events are disabled by setting **DataEventEnabled** to false. This causes subsequent input data to be enqueued while the application processes the current input and associated properties. When the application has finished processing the current input and is ready for more data, it reenables events by setting **DataEventEnabled** to true.

An **ErrorEvent** (or events) is enqueued if an error occurs while gathering or processing input, and is delivered to the application when **DataEventEnabled** is true and other event delivery requirements are met.

The **DataCount** property may be read to obtain the total number of enqueued DataEvents.

All enqueued input may be deleted by calling **clearInput** method. See the **clearInput** method description for more details.

All data properties that are populated as a result of firing a **DataEvent** or **ErrorEvent** can be set back to their default values by calling the **clearInputProperties** method.

The application will be informed about any status change with a **StatusUpdateEvent**, also all corresponding status properties will be updated before event delivery.

Types of voice recognition

Voice recognition is mainly a method of specifying word candidates to be recognized and waiting for those words.

There are the following four types of voice recognition.

Yes/No/Cancel recognition

It listens to the sound of words classified as Yes / No / Cancel defined by the device.

For example, the voice ""OK."" is classified as Yes.

The recognized content is set in the **HearingDataWord** property.

For details, refer to the **startHearingYesNo** method.

Word recognition

The application specifies a list of words and listens for the voice of that word.

The recognized content is set in the **HearingDataWord** property.

For details, refer to the **startHearingWord** method.

[Goto Table 1-173](#)

UPOS Ver1.16 RCSD Specification

Sentence recognition

The application specifies a word and a list of patterns of the sentences using it and awaits the sound of the sentence.

The recognized content is set in the HearingDataWordList property, **HearingDataPattern** property.

For details, see the **startHearingSentence** method.

Free recognition

Voice recognition leave to the device is performed without specifying the word to wait.

[It does not specify waiting words and performs voice recognition entrusted to the device.](#)

The recognized content is set in the **HearingDataWord** property.

For details, see the **startHearingFree** method.

When recognizing voice, the kind of recognition was stored in the **HearingResult** property.

[Goto Table 1-173](#)

Device Sharing

The Voice Recognition is an exclusive-use device, as follows:

- The application must claim the device before enabling it.
- The application must claim and enable the device before accessing some properties or calling methods that update the device.
- See the “Summary” table for precise usage prerequisites.

UPOS Ver1.16 RCSD Specification

Properties (UML attributes)

CapLanguage Property

- Syntax** CapLanguage: *boolean* {read-only, access after open}
- Remarks** If true, the application can change the language. If false, the application cannot change the language.
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.

HearingDataPattern Property

- Syntax** HearingDataPattern: *string* {read-only, access after open-claim-enable}
- Remarks** The pattern ID recognized by the **startHearingSentence** method is set.
This property is set by the **device** control just before the **DataEvent** is **notified** **enqueued**.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
- See Also** **startHearingSentence** Method

[Goto Table 1-174](#)

HearingDataWord Property

- Syntax** HearingDataWord: *string* {read-only, access after open-claim-enable}
- Remarks** The content of voice recognition is set.
This property is set as input data of the following method. To know which method it is for, check the **HearingResult** property.
- | <u>Methods</u> | <u>Meaning</u> |
|---------------------------------|--|
| startHearingYesNo Method | The recognized word is set. |
| startHearingWord Method | Recognized words are set among the word candidates specified by the startHearingWord method. |
| startHearingFree Method | Recognized words and sentences are set.
The alphabet 's uppercase letters, Japanese kanji, hiragana, katakana, etc., the contents to be set varies depending on the device. |
- This property is set by the device control just before the **DataEvent** is **notified** **enqueued**.
- Errors** A **UposException** may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
- See Also** **HearingResult** Property, **startHearingYesNo** Method, **startHearingWord** Method, **startHearingFree** Method

[Goto Table 1-175](#)

UPOS Ver1.16 RCSD Specification
HearingDataWordList Property

Syntax **HearingDataWordList:** *string* {read-only, access after open-claim-enable}

Remarks Comma-separated list of word information recognized by the **startHearingSentence** method.
 Each word information consists of the following information and is shown in the following order separated by a colon (":").

Parameter	Description
<i>WordGoupID</i>	Recognized word group ID
<i>Word</i>	Recognized words. The content defined in the word group is set.

For example, in the **startHearingSentence** method, set candidates as follows,
 Word list: ~~"item:coffee:tea, number:one:two"~~ "item:coffee:tea, count:a:two:three"
~~Sentence pattern "Pattern01: [product] as [number], Pattern02: as [goods] please" When you recognize the word "one coffee." In the pattern "Pattern 01", "coffee" of the word group "product" and "one" of "number" are recognized. When you recognize the word "one coffee." In the pattern "Pattern01", "coffee" of the word group "product" and "one" of "number" are recognized.~~
~~At that time, it looks like the following. "Item: coffee, number: one"~~
 Pattern list: "P1:[count] cup of [item], P2:[item]"
startHearingSentence ("en-US", "item:coffee:tea, count:a:two", "P1:[count] cup of [item],P2:[item]")

If you speak "Give me two cups of coffee", device recognize "Pattern" as "P1" and "WordList" as "item:coffee, count:two".

The properties are set as follows,
 HearingDataPattern="P1";
 HearingDataWordList="item:coffee, count:two";

This property is set by the device control just before the **DataEvent** is enqueued ~~notified~~.

Errors A **UposException** may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.

See Also **startHearingSentence** Method

[Goto Table 1-176](#)

UPOS Ver1.16 RCSD Specification
HearingResult Property

Syntax **HearingResult: int32 {read-only, access after open-claim-enable}**

Remarks A value indicating the voice recognition result is set.
The parameters to be set are as follows.

Value	Meaning
TTS VRCG_HRESULT_YESNO_YES	Voice recognition result of StartHearingYesNo finish running voice recognition methods. Also, Device got an answer that is classified as YES. The recognized content is set in the HearingDataWord finish running voice recognition property.
TTS VRCG_HRESULT_YESNO_NO	Voice recognition result of finish running voice recognition startHearingYesNo method. Also, Device got an answer that is classified as NO. The recognition content is set in the HearingDataWord property.
TTS VRCG_HRESULT_YESNO_CANCEL	Voice recognition result of startHearingYesNo method. Also, Device got responses that are classified as CANCEL. The recognition content is set in the HearingDataWord property.
TTS VRCG_HRESULT_WORD	Recognition result of startHearingWord method. The recognition content is set in the HearingDataWord property.
TTS VRCG_HRESULT_SENTENCE	Recognition result of startHearingSentence method. The recognition content is set in the HearingDataWordList property and HearingDataPattern property.
TTS VRCG_HRESULT_FREE	Recognition result of startHearingFree method. The recognition content is set in the HearingDataWord property.

This property is set by the device control just before the **DataEvent** is ~~notified~~ **enqueued**.

Errors A **UposException** may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.

See Also **HearingDataWord** Property, **HearingDataWordList** Property, **HearingDataPattern** Property, **startHearingYesNo** Method, **startHearingWord** Method, **startHearingSentence** Method, **startHearingFree** Method. [Goto Table 1-177](#)

UPOS Ver1.16 RCSD Specification

HearingStatus Property

Syntax	HearingStatus: int32 {read-only, access after open-claim-enable}												
Remarks	A value indicating the voice recognition status is set. <table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>VRCG#TTS_HSTATUS_NONE</td><td>Voice recognition is not running.</td></tr><tr><td>VRCG#TTS_HSTATUS_YESNO</td><td>Voice recognition by the startHearingYesNo method is in progress.</td></tr><tr><td>VRCG#TTS_HSTATUS_WORD</td><td>Voice recognition by the startHearingWord method is in progress.</td></tr><tr><td>VRCG#TTS_HSTATUS_SENTENCE</td><td>Voice recognition by the startHearingSentence method is in progress.</td></tr><tr><td>VRCG#TTS_HSTATUS_FREE</td><td>Voice recognition by the startHearingFree method is in progress.</td></tr></tbody></table> <p>This property is initialized by the open method. Also, it is set by the device control just before the voice recognition state changes.</p>	Value	Meaning	VRCG#TTS_HSTATUS_NONE	Voice recognition is not running.	VRCG#TTS_HSTATUS_YESNO	Voice recognition by the startHearingYesNo method is in progress.	VRCG#TTS_HSTATUS_WORD	Voice recognition by the startHearingWord method is in progress.	VRCG#TTS_HSTATUS_SENTENCE	Voice recognition by the startHearingSentence method is in progress.	VRCG#TTS_HSTATUS_FREE	Voice recognition by the startHearingFree method is in progress.
Value	Meaning												
VRCG#TTS_HSTATUS_NONE	Voice recognition is not running.												
VRCG#TTS_HSTATUS_YESNO	Voice recognition by the startHearingYesNo method is in progress.												
VRCG#TTS_HSTATUS_WORD	Voice recognition by the startHearingWord method is in progress.												
VRCG#TTS_HSTATUS_SENTENCE	Voice recognition by the startHearingSentence method is in progress.												
VRCG#TTS_HSTATUS_FREE	Voice recognition by the startHearingFree method is in progress.												
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.												
See Also	startHearingYesNo Method, startHearingWord Method, startHearingSentence Method, startHearingFree Method												

[Goto Table 1-178](#)

LanguageList Property

Syntax	LanguageList: string {read-only, access after open}
Remarks	Contains the comma-delimited list of language that are supported by the device. The value representing the language is a value consisting of the language and country code defined in RFC 4664. For example, when the device supports US / English, Japan / Japanese, it will be as follows. "en-US, ja-JP" This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	startHearingYesNo Method, startHearingWord Method, startHearingSentence Method, startHearingFree Method.

Methods (UML operations)

startHearingFree Method

Syntax **startHearingFree (language: string):**
 void {raises-exception, use after open-claim-enable}

Parameter	Description
<i>Language</i>	Specify the language to recognize. Specify one of the values listed in the LanguageList property.

Remarks This method can make a voice recognition from the listed language in the **LanguageList** property. In addition, this method can be called without specifying the word candidate to be recognized from the application, however recognized word depends on the word recognizing device capability. When this method is called, proper values are set in the **HearingDataWord** property, **HearingResult** property and **HearingStatus** property just before the **DataEvent** issuing. ~~Device will start waiting without specifying waiting candidates.~~ This method is executed asynchronously. ~~You can end voice recognition by calling the stopHearing method.~~ Voice recognition ends when **stopHearing** method is called.

Errors A **UposException** may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.

Some possible values of the exception's **ErrorCode** property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified. Or an unsupported language was specified.
E_BUSY	Voice recognition in progress so it cannot be executed.

See Also **HearingDataWord** Property, **HearingResult** Property, **HearingStatus** Property, **LanguageList** Property, **stopHearing** Method.

[Goto Table 1-179](#)

**UPOS Ver1.16 RCSD Specification
startHearingSentence Method**

Syntax `startHearingSentence (language: string, wordList: string,
patternList: string):
void {raises-exception, use after open-claim-enable}`

Parameter	Description
<i>language</i>	Specify the language to recognize. Specify one of the values listed in the LanguageList property.
<i>wordList</i>	Specify word candidates to be waited on in a comma-separated list.
<i>patternList</i>	Specify the sentence pattern information to be waited for in a comma-separated list.

Each word information specified in wordList consists of the following information and is shown in the following order, separated by a colon (":").

Parameter	Description
<i>wordGroupID</i>	ID to identify word list
<i>wordList</i>	A word candidate to be waited for being separated by a colon (":")

For example, to specify word candidates "one" and "two" for word candidate's "coffee" "tea" and word group "number" in the single item group "product", specify as follows. "item:coffee:tea, number:one:two"

Each word information specified in patternList consists of the following information, and it is shown in the following order separated by a colon (":").

Parameter	Description
<i>patternID</i>	ID to identify the pattern
<i>pattern</i>	A sentence pattern to wait. To add the word list specified in wordList to the candidate, enclose the word group ID with "[" and "]". Example: "[word group ID1] [word group ID2] "

~~For example, in wordList, "item: coffee: tea, number: one: two" is specified, and a pattern requesting goods and number such as "Two coffee please" and a pattern requesting goods such as "Coffee please" When defining, specify as follows. "Pattern 01: [Number] [Product] Please, Pattern 02: [Product] please"~~

Example: You can order coffee or tea. You can also specify how many cups you need. If you want to recognize it by voice, do as follows.

Set the **startHearingSentence** method parameter as follows:

`WordList:"item:coffee:tea, count:a:two:three"`

`Coffee, Tea -> item:coffee:tea`

`How many cups -> count:a:two:three`

Invoke the method.

`startHearingSentence ("en-US", "item:coffee:tea,count:a:two", "P1:[count] cup of [item],P2:[item]")`

`HearingStatus=VRCG_HSTATUS_SENTENCE;`

People talk to "Give me two cups of coffee"

Speech recognition is performed, properties are set, and an event is notified.

`HearingResult=VRCG_HRESULT_SENTENCE;`

`HearingDataPattern="P1";`

`HearingDataWordList="item:coffee,count:two";`

`raise DataEvent(0);`

UPOS Ver1.16 RCSD Specification

Remarks ~~Start waiting for sentences defined in wordList and patternList.~~
This method can make a voice recognition from the listed language in the **LanguageList** property. In addition, this method can recognize the words and sentences that are defined in *wordList* and *patternList* as parameter. When this method is called, proper values are set in the **HearingDataWord** property, **HearingResult** property and **HearingStatus** property, just before **DataEvent** issuing. This method is executed asynchronously. Voice recognition ends when **stopHearing** method is called. ~~You can end voice recognition by calling the stopHearing method.~~

Errors A **UposException** may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.

Some possible values of the exception’s **ErrorCode** property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified. Or an unsupported language was specified.
E_BUSY	Voice recognition in progress so it cannot be executed.

See Also **HearingDataWord** Property, **HearingResult** Property, **HearingStatus** Property, **LanguageList** Property, **stopHearing** Method

[Goto Table 1-180](#)

startHearingWord Method

Syntax **startHearingWord** (language: *string*, wordList: *string*):
void {raises-exception, use after open-claim-enable}

Parameter	Description
<i>language</i>	Specify the language to recognize. Specify one of the values listed in the LanguageList property.
<i>wordList</i>	Specify word candidates to be waited on in a comma-separated list. Example: "word1, word2, word3"

Remarks ~~Start waiting for word candidates specified in wordList.~~
This method can make a voice recognition from the listed language in the **LanguageList** property. In addition, this method can recognize the words that are defined in *wordList* as parameter. When this method is called, proper values are set in the **HearingDataWord** property, **HearingResult** property and **HearingStatus** property just before **DataEvent** issuing. This method is executed asynchronously. ~~Application can end voice recognition by calling the stopHearing method.~~ Voice recognition ends when **stopHearing** method is called.

Errors A **UposException** may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.

Some possible values of the exception’s **ErrorCode** property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified. Or an unsupported language was specified.
E_BUSY	Voice recognition in progress so it cannot be executed.

See Also **HearingDataWord** Property, **HearingResult** Property, **HearingStatus** Property, **LanguageList** Property, **stopHearing** Method.

[Goto Table 1-181](#)

UPOS Ver1.16 RCSD Specification
startHearingYesNo Method

Syntax **startHearingYesNo (language: string):**
 void {raises-exception, use after open-claim-enable}

Parameter	Description
------------------	--------------------

<i>language</i>	Specify the language to recognize. Specify one of the values listed in the LanguageList property.
-----------------	--

Remarks ~~Waiting for word candidates corresponding to "Yes" "No" "Cancel" defined by the device is started.~~ This method can make a voice recognition from the listed language in the **LanguageList** property. In addition, this method can recognize the words that are defined in the device as the recognition candidate corresponding to "Yes" "No" "Cancel". When this method is called, proper values are set in the **HearingDataWord** property, **HearingResult** property and **HearingStatus** property, just before **DataEvent** issuing. This method is executed asynchronously. ~~Application can end voice recognition by calling the stopHearing method.~~ Voice recognition ends when **stopHearing** method is called.

Errors A **UposException** may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.

Some possible values of the exception’s **ErrorCode** property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified. Or an unsupported language was specified.
E_BUSY	Voice recognition in progress so it cannot be executed.

See Also **LanguageList** Property, **HearingDataWord** Property, **Hearing Result** Property, **LanguageList** Property, **stopHearing** Method.

[Goto Table 1-182](#)

stopHearing Method

Syntax **stopHearing ():**
 void {raises-exception, use after open-claim-enable}

Remarks **Voice Recognition ends when this property called.**
~~Finish running voice recognition.~~ This method is executed synchronously.

Errors A **UposException** may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.

Some possible values of the exception’s **ErrorCode** property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified. Or an unsupported language was specified.

[Goto Table 1-183](#)

UPOS Ver1.16 RCSD Specification
Events (UML interfaces)

DataEvent

<<event>> upos::events::DataEvent

Status : *int32*{read-only}

Description Notifies the application when data from the Voice Recognition device is available to be read.

Attributes This event contains the following attributes:

Attribute	Type	Description
<i>Status</i>	<i>int32</i>	Set to 0.

Remarks Before this event is delivered, the voice recognition information is enqueued into the area that is indicated by the startHearingXXX kinds of methods.

See Also HearingResult Property, "Events" on page Intro-19, **StartHearingYesNo, StartHearingWord, StartHearingSentence, StartHearingFree** Method , **directIO** method. [Goto Teble 1-184](#)

DirectIOEvent

<<event>> upos::events::DirectIOEvent

EventNumber : *int32* {read-only}

Data : *int32* {read-write}

Obj : *object* {read-write}

Description Provides Service information directly to the application. This event provides a means for a vendor-specific Voice Recognition Service to provide events to the application that are not otherwise supported by the device control.

Attributes This event contains the following attributes:

Attribute	Type	Description
<i>EventNumber</i>	<i>int32</i>	Event number whose specific values are assigned by the Service.
<i>Data</i>	<i>int32</i>	Additional numeric data. Specific values vary by the <i>EventNumber</i> and the Service. This attribute is settable.
<i>Obj</i>	<i>object</i>	Additional data whose usage varies by the <i>EventNumber</i> and the Service. This attribute is settable.

Remarks This event is to be used only for those types of vendor specific functions that are not otherwise described.

Use of this event may restrict the application program programform being used with other vendor's devices which may not have any knowledge of the Service's need for this event.

See Also "Events" on page Intro-19, **directIO** method.

[Goto Table 1-185](#)

UPOS Ver1.16 RCSD Specification

ErrorEvent

<<event>> upos::events:: ErrorEvent
Errorcode : int32{read-write}
ErrorcodeExtended : int32{read-write}
ErrorLocus : int32{read-write}
ErrorResponse : int32{read-write}

Description Notifies the application that a Voice Recognition Device error has been detected and suitable response by the application is necessary to process the error condition.

Attributes This event contains the following attributes:

Attributes	Type	Description
<i>ErrorCode</i>	<i>int32</i>	Error code causing the error event. See a list of Error Codes on page 20.
<i>ErrorCodeExtended</i>	<i>int32</i>	Extended Error code causing the error event. If <i>ErrorCode</i> is E_EXTENDED, then see values below. Otherwise, it may contain a Service-specific value.
<i>ErrorLocus</i>	<i>int32</i>	Location of the error. If EL_OUTPUT is specified. An error occurred during asynchronous action.
<i>ErrorResponse</i>	<i>int32</i>	Error response, whose default value may be overridden by the application (i.e., this attribute is settable). See values below.

The *ErrorLocus* attribute has one of the following values:

Value	Meaning
EL_OUTPUT	Error occurred while processing asynchronous output.
EL_INPUT	Error occurred while gathering or processing event-driven input. No previously buffered input data is available.
EL_INPUT_DATA	Error occurred while gathering or processing event-driven input, and some previously buffered data is available.

The application's error event handler can set the *ErrorResponse* attribute to one of the following values:

Value	Meaning
ER_RETRY	Retry sending the data. The error state is exited. May be valid for some input devices when the locus is EL_INPUT, in which case the input is re-tried, and the error state is exited. Typically, valid for asynchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited. This is the default response when the locus is EL_OUTPUT.
ER_CLEAR	Valid for all loci: EL_INPUT, EL_INPUT_DATA, and EL_OUTPUT. Clear all buffered input or output data (including all asynchronous output). The error state is exited. This is the default response when the locus is EL_INPUT.

UPOS Ver1.16 RCSD Specification

ER_CONTINUEINPUT

Only valid when the locus is EL_INPUT_DATA. Acknowledges that a data error has occurred and directs the Device to continue input processing. The Device remains in the error state and will deliver additional **DataEvents** as directed by the **DataEventEnabled** property. When all input has been delivered and **DataEventEnabled** is again set to true, then another **ErrorEvent** is delivered with locus EL_INPUT. This is the default response when the locus is EL_INPUT_DATA.

Remarks This event is enqueued when an error is detected and the Device's **State** transitions into the error state. Input error events are not delivered until **DataEventEnabled** is true, so that proper application sequencing occurs.

Unlike a **DataEvent**, the Device does not disable further **DataEvents** or input **ErrorEvents**; it leaves the **DataEventEnabled** property value at true. Note that the application may set **DataEventEnabled** to false within its event handler if subsequent input events need to be disabled for a period of time.

See Also "Device Input Model" on page Intro-22, "Error Handling" on page Intro-23, "Device Output Models" on page Intro-25. [Goto Table 1-186](#)

StatusUpdateEvent

<<event>> **upos::events:: StatusUpdateEvent**
Status : *int32* {read-only}

Description *Notifies the application that there is a change in the power status or a status of the Voice Recognition device.*

Attributes This event contains the following attribute:

Attributes	Type	Description
------------	------	-------------

Status	int32	Indicates a change in the power status of the unit.
--------	-------	---

Note that Release 1.3 added Power State Reporting with additional Power reporting StatusUpdateEvent values.

The Update Firmware capability added additional Status values for communicating the status/progress of an asynchronous update firmware process. See "StatusUpdateEvent" description on page 1-34.

Value	Meaning
-------	---------

VRCG_SUE_START_HEARING_FREE	It will be notified when hearing free starts.
-----------------------------	---

VRCG_SUE_START_HEARING_SENTENCE	It will be notified when hearing sentence starts.
---------------------------------	---

VRCG_SUE_START_HEARING_WORD	It will be notified when hearing word starts.
-----------------------------	---

VRCG_SUE_START_HEARING_YESNO	It will be notified when hearing yesno starts.
------------------------------	--

VRCG_SUE_STOP_HEARING	It will be notified when hearing stops.
-----------------------	---

Remarks Enqueued when the Voice Recognition Device detects a power state change or a status change.

See Also "Events" on page Intro-19.

[Goto Table 1-187](#)

Sound Player

This Chapter defines the Sound Player device category.

Summary

Properties (UML attributes)

<i>Common</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
AutoDisable:	<i>boolean</i>	{read-write}	1.16	Not supported
CapCompareFirmwareVersion:	<i>boolean</i>	{read-only}	1.16	open
CapPowerReporting:	<i>int32</i>	{read-only}	1.16	open
CapStatisticsReporting:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateFirmware:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateStatistics:	<i>boolean</i>	{read-only}	1.16	open
CheckHealthText:	<i>string</i>	{read-only}	1.16	open
Claimed:	<i>boolean</i>	{read-only}	1.16	open
DataCount:	<i>int32</i>	{read-only}	1.16	Not supported open
DataEventEnabled:	<i>boolean</i>	{read-write}	1.16	Not supported open
DeviceEnabled:	<i>boolean</i>	{read-write}	1.16	open & claim
FreezeEvents:	<i>boolean</i>	{read-write}	1.16	open
OutputID:	<i>int32</i>	{read-only}	1.16	open
PowerNotify:	<i>int32</i>	{read-write}	1.16	open
PowerState:	<i>int32</i>	{read-only}	1.16	open
State:	<i>int32</i>	{read-only}	1.16	--
DeviceControlDescription:	<i>string</i>	{read-only}	1.16	-
DeviceControlVersion:	<i>int32</i>	{read-only}	1.16	-
DeviceServiceDescription:	<i>string</i>	{read-only}	1.16	open
DeviceServiceVersion:	<i>int32</i>	{read-only}	1.16	open
PhysicalDeviceDescription:	<i>string</i>	{read-only}	1.16	open
PhysicalDeviceName:	<i>string</i>	{read-only}	1.16	open

[Goto Table 1-188](#)

UPOS Ver1.16 RCSD Specification

Properties (Continued)

<i>Specific</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
CapAssociatedHardTotalsDevice	string	{read-write}	1.16	open
CapMultiPlay:	boolean	{read-only}	1.16	open
CapSoundTypeList:	string	{read-only}	1.16	open
CapStorage	int32	{read-only}	1.16	open
CapVolume:	boolean	{read-only}	1.16	open
DeviceSoundList:	string	{read-only}	1.16	open
OutputIDList:	string	{read-only}	1.16	open, claim & enable
Storage	int32	{read-write}	1.16	open, claim & enable
Volume:	int32	{read-write}	1.16	open, claim & enable

Methods (UML operations)

Common

<i>Name</i>	<i>Version</i>
open (logicalDeviceName: string): void {raises-exception}	1.16
close (): void {raises-exception, use after open}	1.16
claim (timeout: int32): void {raises-exception, use after open}	1.16
release (): void {raises-exception, use after open, claim}	1.16
checkHealth (level: int32): void {raises-exception, use after open, enable}	1.16
clearInput (): void { }	Not supported 1.16
clearInputProperties (): void { }	Not supported 1.16
clearOutput (): void { }	Not supported
directIO (command: int32, inout data: int32, inout obj: object): void {raises-exception, use after open}	1.16
compareFirmwareVersion (firmwareFileName: string, out result: int32): void {raises-exception, use after open, claim, enable}	1.16
resetStatistics (statisticsBuffer: string): void {raises-exception, use after open, claim, enable}	1.16
retrieveStatistics (inout statisticsBuffer: string): void {raises-exception, use after open, claim, enable}	1.16

[Goto Table 1-189](#)
[Goto Table1-295](#)

UPOS Ver1.16 RCSD Specification

Methods (UML operations)(continued)

Common

<i>Name</i>	<i>Version</i>
updateFirmware (firmwareFileName: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16
updateStatistics (statisticsBuffer: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16

Specific

<i>Name</i>	<i>Version</i>
playSound (fileName: <i>string</i> , loop: <i>boolean</i>): void { raises-exception, use after open, claim, enable}	1.16
stopSound (outputID: <i>int32</i>): void {raises-exception, use after open, claim, enable}	1.16

Events (UML interfaces)

<i>Name</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>
upos::events::DataEvent		<i>Not supported</i>	1.16
upos::events::DirectIOEvent			1.16
EventNumber:	<i>int32</i>	{read-only}	
Data:	<i>int32</i>	{read-write}	
Obj:	<i>object</i>	{read-write}	
upos::events::ErrorEvent			1.16
ErrorCode:	<i>int32</i>	{read-only}	
ErrorCodeExtended:	<i>int32</i>	{read-only}	
ErrorLocus:	<i>int32</i>	{read-only}	
ErrorResponse:	<i>int32</i>	{read-write}	
upos::events::OutputCompleteEvent			1.16
OutputID:	<i>int32</i>	{read-only}	
upos::events::StatusUpdateEvent			1.16
Status:	<i>int32</i>	{read-only}	
upos::events::TransitionEvent		<i>Not supported</i>	1.16

[Goto Table 1-190](#)

UPOS Ver1.16 RCSD Specification

General Information

The Sound Player programmatic name is "Sound Player".

Capabilities

The Sound Player has the following capability:

- Play audio file.

Sound Player Class Diagram

The following diagram shows the relationships between the Sound player classes.

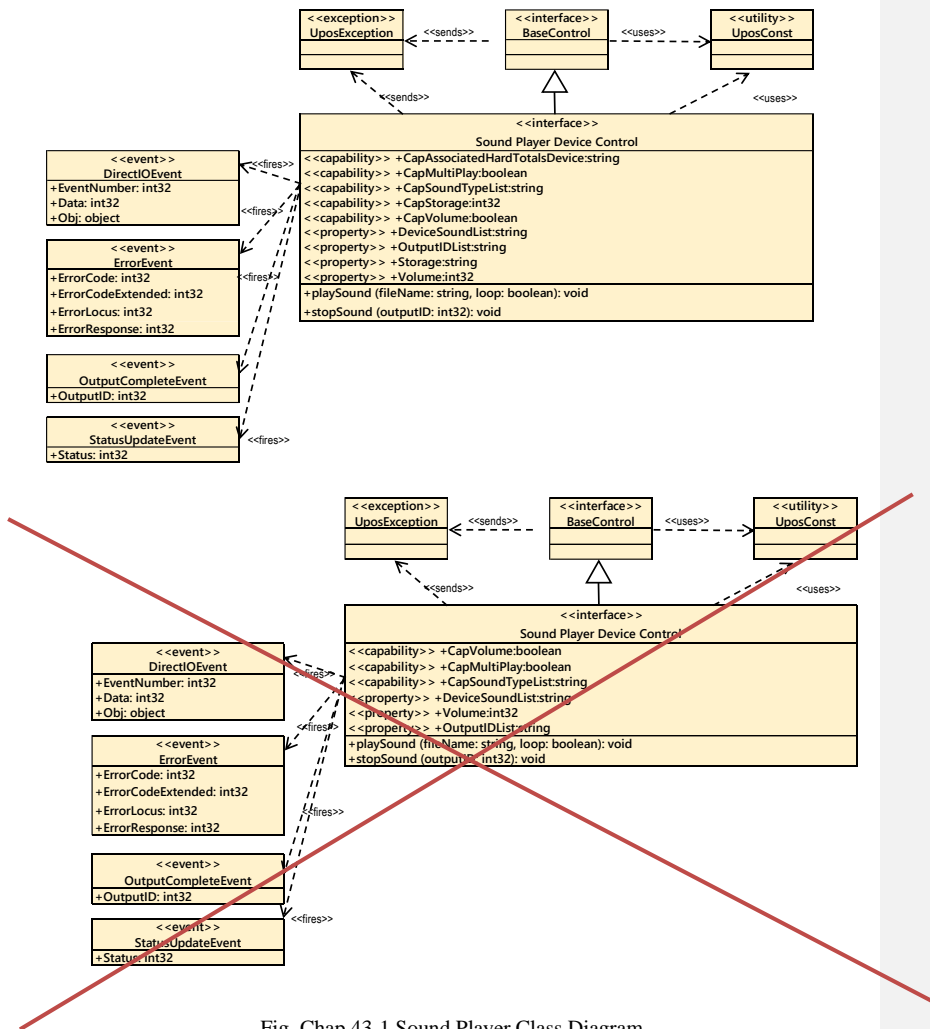


Fig. Chap.43-1 Sound Player Class Diagram

[Goto Table2-6](#)

UPOS Ver1.16 RCSD Specification

Model

The Sound Player follows the general device behavior model for asynchronous output devices:

- The Device validates the method parameters and produces an error condition immediately if necessary. If the validation is successful, the Device does the following:
 - ~~• The application calls a startSound method to start playing sound. The Device validates the method parameters and produces an error condition immediately if necessary. If the validation is successful, the Device does the following:~~
 - ~~• "1. Buffers the request in program memory, for delivery to the Physical Device as soon as the Physical Device can receive and process it.~~
 - ~~• 2. Sets the OutputID property to a unique integer identifier for this request.~~
 - ~~• 3. Returns as soon as possible."~~
 - ~~• When the Device successfully completes a request, an OutputCompleteEvent is enqueued for delivery to the application. A property of this event contains the output ID of the completed request. The application should compare the returned OutputCompleteEvent property OutputID value with the OutputID value set by the asynchronous process method call used to send the data in order to track what data has been successfully sent to the device.~~
- Audio files will be played sequentially. When **playSound** method is called, device starts the playing sound that is specified by the method parameters and the requested sound file data placed in a queue and corresponding OutputID is stored at **OutputID** property and added to the **OutputIDList** property as a listed value. And sets the **OutputID** property to a unique integer identifier for this request.
- When the sound playing starts **StatusUpdateEvent** is evoked as the value of SPLY_SUE_START_PLAY_SOUND.
When the sound playing is finished an **OutputCompleteEvent** is enqueued for the delivery to the application and corresponding OutputID is stored in **OutputID** property. At the same time, **StatusUpdateEvent** is evoked as the value of SPLY_SUE_STOP_PLAY_SOUND. The application should compare the returned **OutputCompleteEvent** property **OutputID** value with the **OutputID** value set by the asynchronous process method call used to send the data in order to track what data has been successfully sent to the device.
- When **stopSound** method is called, device stop the playing sound according to the OutputID property value and the current playing sound is terminated and enqueued sound file data is cleared. After this method is executed, corresponding **OutputID** property and **OutputIDList** values are not changed. No **OutputCompleteEvent** is fired and only **StatusUpdateEvent** will be evoked the value of SPLY_SUE_STOP_PLAY_SOUND.
- If an error occurs while processing a request, an **ErrorEvent** is enqueued which will be delivered to the application after the events already enqueued, including **OutputCompleteEvent**. No further asynchronous output will occur until the event has been delivered to the application. If the response is ER_CLEAR, then outstanding asynchronous output is cleared. If the response is ER_RETRY, then output is retried;

[Goto Table 1-191](#)

UPOS Ver1.16 RCSD Specification

note that if several outputs were simultaneously in progress at the time that the error was detected, then the Service may need to retry all of these outputs.

- Asynchronous output is always performed on a first-in first-out basis. If the device supports concurrent playback, the request will be executed simultaneously. To check if the device supports simultaneous playback, check the **CapMultiPlay** property.
- If the request is terminated before completion, due to reasons such as the application calling the **clearOutput** method, then no **OutputCompleteEvent** is delivered.
- Application can also delete the output individually by calling the **stopSound** method. Also, in this case **OutputCompleteEvent** will not be notified."
- The **CapSoundTypeList** property lists audio file types that the device can play.
- The application will be informed about any status change with a **StatusUpdateEvent**, also all corresponding status properties will be updated before event delivery.
- ~~Applications need to support "hard total" services as audio files played with the startSound method must be placed in the area managed by the "hard total" service.~~
- If device supports either or both of Hard Totals devices and the host file system, the application should set the **Storage** property accordingly to tell where to access the data file.
- If device needs to be able to access the audio files played with **playSound** method from a Hard Totals device, the **CapAssociatedHardTotalsDevice** property holds the open name of the associated Hard Totals device.

[Goto Table 1-191](#)

Device Sharing

The Sound Player is an exclusive-use device, as follows:

- The application must claim the device before enabling it.
- The application must claim and enable the device before accessing some properties or calling methods that update the device.
- See the "Summary" table for precise usage prerequisites.

UPOS Ver1.16 RCSD Specification
Properties(UML attributes)

CapAssociatedHardTotalsDevice Property

Syntax CapAssociatedHardTotalsDevice : *string* {read-write, access after open}

Remarks Holds the open name of the associated Hard Totals device if the device is able to write to such devices which is the case if **CapStorage** is either SPLY_CST_ALL or SPLY_CST_HARDTOTALS_ONLY. If **CapStorage** is SPLY_CST_HOST_ONLY this property value must be the empty string.

Errors UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.

See Also **CapStorage** Property [Goto Table 1-192](#)

CapMultiPlay Property

Syntax CapMultiPlay : *boolean* {read-only, access after open}

Remarks If true, the application can play sound simultaneously.
If false, the application cannot play sound simultaneously.
This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.

See Also **playSound** Method.

CapSoundTypeList Property

Syntax CapSoundTypeList : *string* {read-only, access after open}

Remarks Contains the comma-delimited list of file type that is supported by the device.
For example, if the device only supports WAV and OGG, then this property should be set to "WAV, OGG". This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.

See Also **playSound** Method

UPOS Ver1.16 RCSD Specification

CapStorage Property

Syntax **CapStorage: *int32* {read-only, access after open}**

Remarks This is an enumeration and announces where the device is able to write the recorded sound data file to.
It holds one of the following values.

<u>Value</u>	<u>Meaning</u>
SPLY_CST_HARDTOTALS_ONLY	Only an associate Hard Totals device is supported.
SPLY_CST_HOST_ONLY	Only the host's file system is supported.
SPLY_CST_ALL	Both, the associated Hard Totals device and the host's file system is supported.

This property is initialized by the **open** method.

If a Hard Totals device is supported the Storage, the property value should be SPLY_CST_HARDTOTALS_ONLY or SPLY_CST_ALL and the property **CapAssociatedHardTotalsDevice** holds the open name of the associated **Hard Totals** device. [Goto Table 1-193](#)

CapVolume Property

Syntax **CapVolume: *boolean* {read-only, access after open-~~claim~~}**

Remarks If true, the application can change the volume during playback.
If false, the application cannot change the volume during playback.
This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed. For further information, see "**Errors**" on page Intro-20.

See Also **Volume** Property. [Goto Table 1-194](#)

DeviceSoundList Property

Syntax **DeviceSoundList : *string* {read-only, access after open}**

Remarks Contains the comma-delimited list of device sound ID that is supported by the device. This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed. For further information, see "**Errors**" on page Intro-20.

See Also **playSound** Method

OutputIDList Property

Syntax **OutputIDList : *string* {read-only, access after open, claim-**enable**}**

Remarks Contains the comma-delimited list of OutputID that is output by the **playSound** method. This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed. For further information, see "**Errors**" on page Intro-20.

See Also **playSound** Method

[Goto Table 1-195](#)

UPOS Ver1.16 RCSD Specification

Storage Property

Syntax Storage: *int32* {read-write, access after open-claim-enable}

Remarks It holds one of the following values.

<u>Value</u>	<u>Meaning</u>
SPLY_ST_HARDTOTALS	The encoded data file is written to the associated Hard Totals device. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device.
SPLY_ST_HOST	The encoded data file is written to the host's file system.
SPLY_ST_HOST_HARDTOTALS	The encoded data file is written to the associated Hard Totals device and host's file system. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device.

This property is initialized by the **open** method according to the value hold by **CapStorage**. If **CapStorage** has the value SPLY_CST_ALL, it is initialized to SPLY_ST_HOST_HARDTOTALS.

Errors UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.

See Also [CapStorage Property](#)

[Goto Table 1-196](#)

Volume Property

Syntax Volume : *int32* {read-write, access after open-claim-enable}

Remarks Holds the volume at playing sound.

Legal values range from zero through 100.

This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

See Also [playSound Method](#)

[Goto Table 1-197](#)

UPOS Ver1.16 RCSD Specification

Methods (UML operations)

playSound Method

Syntax **playSound** (fileName : *string*, loop : *boolean*):
 void{raises-exception, use after open-claim-enable}

<u>Parameter</u>	<u>Description</u>
<i>fileName</i>	Specifies the file name of audio file. Or, specifies one of the sound ID defined by DeviceSoundList .
<i>loop</i>	When true is specified, loop playback is performed, and if false is specified, loop playback will not be performed.

Remarks Play audio file specified by fileName or device definition sound.

Audio files **might** ~~must~~ be located in the area managed by "**Hard Totals**" service.

This method will be performed asynchronously. To stop playback, call the **stopSound** method.

Errors A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20. Some possible values of the exception's ErrorCode property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified. Or an unsupported sound file was specified.
E_NOEXIST	File does not exist.

See Also **CapSoundType** Property, **DeviceSoundList** Property, **stopSound** Method

[Goto Table 1-198](#)

stopSound Method

Syntax **stopSound**(outputID: *int32*):
 void{raises-exception, use after open-claim-enable}

<u>Parameter</u>	<u>Description</u>
<i>outputID</i>	Specify the outputID of the sound to stop.

Remarks Terminates specified audio playback [according to the OutputID property value](#).

Errors A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20. Some possible values of the exception's ErrorCode property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	The specified sound is not being played.

See Also **OutputID** Property, **startSound** Method

[Goto Table 1-199](#)

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

DirectIOEvent

<<event>> **upos::events::DirectIOEvent**

EventNumber : *int32* {read-only}
Data : *int32* {read-write}
Obj : *object* {read-write}

Description Provides Service information directly to the application. This event provides a means for a vendor-specific Sound Player Service to provide events to the application that are not otherwise supported by the device control.

Attributes This event contains the following attributes:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>EventNumber</i>	<i>int32</i>	Event number whose specific values are assigned by the Service.
<i>Data</i>	<i>int32</i>	Additional numeric data. Specific values vary by the <i>EventNumber</i> and the Service. This attribute is settable.
<i>Obj</i>	<i>object</i>	Additional data whose usage varies by the <i>EventNumber</i> and the Service. This attribute is settable.

Remarks This event is to be used only for those types of vendor specific functions that are not otherwise described.

Use of this event may restrict the application program programform being used with other vendor's devices which may not have any knowledge of the Service's need for this event.

See Also "Events" on page Intro-19, **directIO** method

[Goto Table 1-200](#)

UPOS Ver1.16 RCSD Specification

ErrorEvent

```
<<event>> upos::events:: ErrorEvent
  ErrorCode          : int32{read-write}
  ErrorCodeExtended  : int32{read-write}
  ErrorLocus         : int32{read-write}
  ErrorResponse      : int32{read-write}
```

Description Notifies the application that a Sound Player Device error has been detected and suitable response by the application is necessary to process the error condition.

Attributes This event contains the following attributes:

Attributes	Type	Description
<i>ErrorCode</i>	<i>int32</i>	Error code causing the error event. See a list of Error Codes on page 20.
<i>ErrorCodeExtended</i>	<i>int32</i>	Extended Error code causing the error event. If <i>ErrorCode</i> is E_EXTENDED, then see values below. Otherwise, it may contain a Service-specific value.
<i>ErrorLocus</i>	<i>int32</i>	Location of the error. If EL_OUTPUT is specified. An error occurred during asynchronous action.
<i>ErrorResponse</i>	<i>int32</i>	Error response, whose default value may be overridden by the application (i.e., this attribute is settable). See values below.

If *ErrorCode* is E_EXTENDED, then *ErrorCodeExtended* has one of the following values:

Value	Meaning
ESPLY_NOROOM	The encoded data storage area does not have enough room to store. The <i>ErrorLocus</i> attribute has one of the following values:

Value	Meaning
EL_OUTPUT	Error occurred while processing asynchronous output.

The application's error event handler can set the *ErrorResponse* attribute to one of the following values:

Value	Meaning
ER_RETRY	Retry sending the data. The error state is exited. May be valid for some input devices when the locus is EL_INPUT, in which case the input is re-tried, and the error state is exited. Typically, valid for asynchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited. This is the default response when the locus is EL_OUTPUT.
ER_CLEAR	Valid for all loci: EL_INPUT, EL_INPUT_DATA, and EL_OUTPUT. Clear all buffered input or output data (including all asynchronous output). The error state is exited. This is the default response when the locus is EL_INPUT.

Remarks This event is enqueued when an error is detected and the Device's **State** transitions into the error state.

See Also "Error Handling" on page Intro-23, "Device Output Models" on page Intro-25. [Goto Table 1-201](#)

UPOS Ver1.16 RCSD Specification

OutputCompleteEvent

<<event>> **upos::events::OutputCompleteEvent**
OutputID : *int32*{read-only}

Description Notify the application that the queued output request associated with the *outputID* property has completed successfully.

Attributes This event contains the following attributes:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>OutputID</i>	<i>int32</i>	The ID number of the asynchronous output request that is complete.

Remarks This event is enqueued after the request's data has been both sent, and the Service has confirmation that it was processed by the device successfully.

See Also "Device Output Models" on page Intro-25

[Goto Table 1-202](#)

StatusUpdateEvent

<<event>> **upos::events::StatusUpdateEvent**
Status : *int32* {read-only}

Description *Notifies the application that there is an operation status change or a status of the sound player device.*

Attributes This event contains the following attribute:

<u>Attributes</u>	<u>Type</u>	<u>Description</u>
<i>Status</i>	<i>int32</i>	Indicates a change of operation status of sound player device.

*Note that Release 1.3 added Power State Reporting with additional Power reporting **StatusUpdateEvent** values.*

The Update Firmware capability added additional *Status* values for communicating the status/progress of an asynchronous update firmware process.

See "**StatusUpdateEvent**" description on page 1-34.

<u>Value</u>	<u>Meaning</u>
SPLY_SUE_START_PLAY_SOUND	It will be notified when sound playing start.

SPLY_SUE_STOP_PLAY_SOUND	It will be notified when sound playing stop.
--------------------------	--

Remarks Enqueued when the Sound Player Device detects a power state change or a status change.

See Also "Events" on page Intro-19.

[Goto Table 1-203](#)

Speech Synthesis

This Chapter defines the Speech Synthesis device category.

Summary

Properties (UML attributes)

<i>Common</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
AutoDisable:	<i>boolean</i>	{read-write}	1.16	Not S upported
CapCompareFirmwareVersion:	<i>boolean</i>	{read-only}	1.16	open
CapPowerReporting:	<i>int32</i>	{read-only}	1.16	open
CapStatisticsReporting:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateFirmware:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateStatistics:	<i>boolean</i>	{read-only}	1.16	open
CheckHealthText:	<i>string</i>	{read-only}	1.16	open
Claimed:	<i>boolean</i>	{read-only}	1.16	open
DataCount:	<i>int32</i>	{read-only}	1.16	Not S upported
DataEventEnabled:	<i>boolean</i>	{read-write}	1.16	Not S upported
DeviceEnabled:	<i>boolean</i>	{read-write}	1.16	open & claim
FreezeEvents:	<i>boolean</i>	{read-write}	1.16	open
OutputID:	<i>int32</i>	{read-only}	1.16	open
PowerNotify:	<i>int32</i>	{read-write}	1.16	open
PowerState:	<i>int32</i>	{read-only}	1.16	open
State:	<i>int32</i>	{read-only}	1.16	--
DeviceControlDescription:	<i>string</i>	{read-only}	1.16	--
DeviceControlVersion:	<i>int32</i>	{read-only}	1.16	--
DeviceServiceDescription:	<i>string</i>	{read-only}	1.16	open
DeviceServiceVersion:	<i>int32</i>	{read-only}	1.16	open
PhysicalDeviceDescription:	<i>string</i>	{read-only}	1.16	open
PhysicalDeviceName:	<i>string</i>	{read-only}	1.16	open

[Goto Table 1-205](#)

UPOS Ver1.16 RCSD Specification

Properties (Continued)

<i>Specific</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
CapLanguage:	<i>boolean</i>	{read-only}	1.16	open
CapPitch:	<i>boolean</i>	{read-only}	1.16	open
CapSpeed:	<i>boolean</i>	{read-only}	1.16	open
CapVoice:	<i>boolean</i>	{read-only}	1.16	open
CapVolume:	<i>boolean</i>	{read-only}	1.16	open
Language:	<i>string</i>	{read-write}	1.16	open, claim & enable
LanguageList:	<i>string</i>	{read-only}	1.16	open
OutputIDList:	<i>string</i>	{read-only}	1.16	open, claim & enable
Pitch:	<i>int32</i>	{read-write}	1.16	open, claim & enable
Speed:	<i>int32</i>	{read-write}	1.16	open, claim & enable
Voice:	<i>string</i>	{read-write}	1.16	open, claim & enable
VoiceList:	<i>string</i>	{read-only}	1.16	open
Volume:	<i>int32</i>	{read-write}	1.16	open, claim & enable

Methods (UML operations)

Common

<i>Name</i>	<i>Version</i>
open (<i>logicalDeviceName: string</i>): void {raises-exception}	1.16
close (): void {raises-exception, use after open}	1.16
claim (<i>timeout: int32</i>): void {raises-exception, use after open}	1.16
release (): void {raises-exception, use after open, claim}	1.16
checkHealth (<i>level: int32</i>): void {raises-exception, use after open, enable}	1.16
clearInput (): void { }	Not supported 1.16
clearInputProperties (): void { }	Not supported 1.16

[Goto Table 1-296](#)

UPOS Ver1.16 RCSD Specification

Methods (UML operations)(continued)

clearOutput ():
void {} 1.16

Common

<i>Name</i>	<i>Version</i>
compareFirmwareVersion (firmwareFileName: <i>string</i> , out result: <i>int32</i>): void {raises-exception, use after open, claim, enable}	1.16
directIO (command: <i>int32</i> , inout data: <i>int32</i> , inout obj: <i>object</i>): void {raises-exception, use after open}	1.16
resetStatistics (statisticsBuffer: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16
retrieveStatistics (inout statisticsBuffer: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16
updateFirmware (firmwareFileName: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16
updateStatistics (statisticsBuffer: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16

Specific

<i>Name</i>	
speak (text: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16
speakImmediate (text: <i>string</i>): void {raises-exception, use after open, claim, enable}	1.16
stopCurrentSpeaking (): void {raises-exception, use after open, claim, enable}	1.16
stopSpeaking (outputID: <i>int32</i>): void {raises-exception, use after open, claim, enable}	1.16

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

<i>Name</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>
upos::events::DataEvent		<i>Not supported</i>	
upos::events::DirectIOEvent			1.16
EventNumber:	<i>int32</i>	{read-only}	
Data:	<i>int32</i>	{read-write}	
Obj:	<i>object</i>	{read-write}	
upos::events::ErrorEvent			1.16
ErrorCode:	<i>int32</i>	{read-only}	
ErrorCodeExtended:	<i>int32</i>	{read-only}	
ErrorLocus:	<i>int32</i>	{read-only}	
*pErrorResponse:	<i>int32</i>	{read-write}	
upos::events::OutputCompleteEvent			1.16
OutputID:	<i>int32</i>	{read-only}	
upos::events::StatusUpdateEvent			1.16
Status:	<i>int32</i>	{read-only}	
upos::events::TransitionEvent		<i>Not supported</i>	

[Goto Table 1-206](#)
[Goto Table 1-207](#)

UPOS Ver1.16 RCSD Specification

General Information

The Speech Synthesis programmatic name is "Speech Synthesis".

Capabilities

The Speech Synthesis has the following capability:

- Convert text to speech and ~~read~~ read it aloud.

Speech Synthesis Class Diagram

The following diagram shows the relationships between the Speech Synthesis classes.

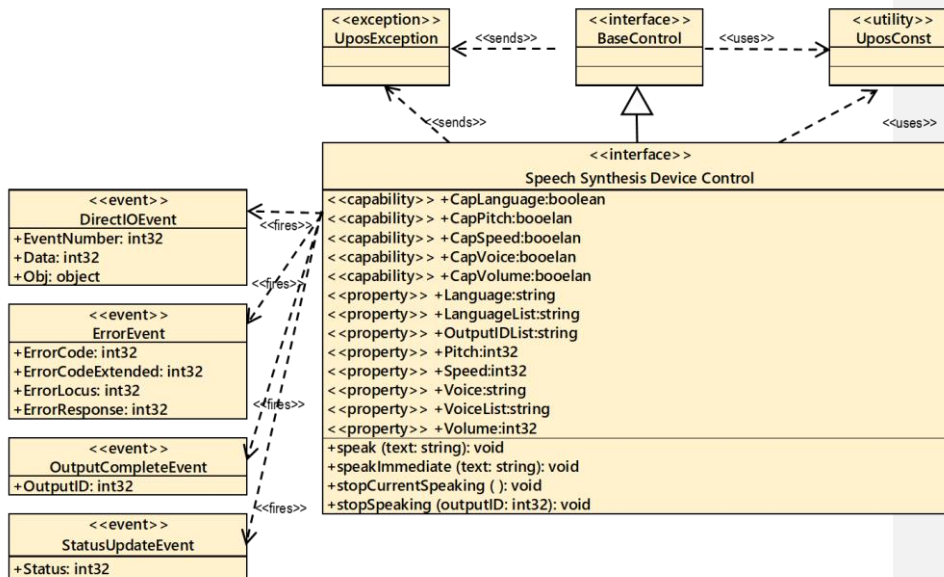


Fig. Chap. 44-1 Speech Synthesis Class Diagram

UPOS Ver1.16 RCSD Specification Model

The Speech Synthesis follows the general device behavior model for ~~asynchronous output devices~~: output devices with some enhancements.

The application calls a **speak** method or **speakImmediate** method to speech.

The **speak** method acts to start speaking from the words specified by text, while the **speakImmediate** method ends immediately previous **speak** method, and starts speaking the word specified by text asynchronously and immediately.

~~The device validates the method parameters and produces an error condition immediately if necessary. If the validation is successful, the device does the following:~~

- ~~1. Buffers the request in program memory, for delivery to the physical device as soon as the physical device can receive and process it.~~
- ~~2. Sets the **OutputID** property to a unique integer identifier for this request.~~
- ~~3. Returns as soon as possible.~~

~~When the device successfully completes a request, an **OutputCompleteEvent** is enqueued for delivery to the application. A property of this event contains the **output ID** of the completed request. The application should compare the returned **OutputCompleteEvent** property's **OutputID** value with the **OutputID** value set by the asynchronous process method call used to send the data in order to track what data has been successfully sent to the device.~~

When **speak** or **speakImmediate** method is called device start the speaking based on the setting value of **Language**, **Volume**, **Pitch** and **Speed** properties. And requested utterance written by text data placed in a queue and corresponding **OutputID** is stored at **OutputID** property and added to the **OutputIDList** property as listed value. And sets the **OutputID** property to a unique integer identifier for this request.

When an utterance of **speak** method or **speakImmediate** method starts, **StatusUpdateEvent** is evoked as the value of SPSY_SUE_START_SPEAK. When the utterance is finished an **OutputCompleteEvent** is enqueued for the delivery to the application and corresponding **OutputID** is stored in **OutputID** property. At the same time **StatusUpdateEvent** is evoked as the value of SPSY_SUE_STOP_SPEAK. The application should compare the returned **OutputCompleteEvent** property **OutputID** value with **OutputID** value set by the asynchronous process method call used to send the data in order to track what data has been successfully sent to the device

When **speakImmediate** method is called during the utterance of **speak** method or **speakImmediate** method call, utterance will be stopped immediately. And **StatusUpdateEvent** is evoked as the value of SPSY_SUE_STOP_SPEAK. However, **OutputCompleteEvent** is not fired. And current **speak** method or **speakImmediate** method corresponding **OutputID** property and **OutputIDList** property values are not changed.

When **stopCurrentSpeaking** method is called, current utterance generated by **speak** method or **speakImmediate** method will be stopped and **StatusUpdateEvent** is evoked as the value of SPSY_SUE_STOP_SPEAK. And no **OutputCompleteEvent** is fired. And current **speak** method or **speakImmediate** method corresponding **OutputID** property and **OutputIDList** property values are not changed.

When **stopSpeaking** method is called, specified **OutputID** valued utterance is stopped and deleted. And **OutputID** property value in the **OutputIDList** property is eliminated.

When utterance is stopped **StatusUpdateEvent** is evoked as the value of SPSY_SUE_STOP_SPEAK. And no **OutputCompleteEvent** is fired.

If an error occurs while processing a request, an **ErrorEvent** is enqueued which will be delivered to the application after the events already enqueued, including **OutputCompleteEvent**. No further asynchronous output will occur until the event has been delivered to the application. If the response is ER_CLEAR, then outstanding

[Goto Table 1-207](#)

UPOS Ver1.16 RCSD Specification

asynchronous output is cleared. If the response is ER_RETRY, then output is retried; note that if several outputs were simultaneously in progress at the time that the error was detected, then the service may need to retry all of these outputs.

Asynchronous output is always performed on a first-in first-out basis.

If the request is terminated before completion, due to reasons such as the application calling the **clearOutput** method, then no **OutputCompleteEvent** is delivered.

~~Application can also delete the output individually by calling the **stopCurrentSpeaking**, **stopSpeaking** method. Also in this case **OutputCompleteEvent** will not be notified.~~

The application will be informed about any status change with a **StatusUpdateEvent**, also all corresponding status properties will be updated before event delivery.

Device Sharing

[Goto Table 1-207](#)

The Speech Synthesis is an exclusive-use device, as follows:

- The application must claim the device before enabling it.
- The application must claim and enable the device before accessing some properties or calling methods that update the device.
- See the “Summary” table for precise usage prerequisites.

UPOS Ver1.16 RCSD Specification

Properties (UML attributes)

CapLanguage Property

Syntax	CapLanguage: <i>boolean</i> {read-only, access after open}
Remarks	If true, the application can change the language. If false, the application cannot change the language. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	Language Property

CapPitch Property

Syntax	CapPitch: <i>boolean</i> {read-only, access after open}
Remarks	If true, the application can change the pitch. If false, the application cannot change the pitch. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	Pitch Property

CapSpeed Property

Syntax	CapSpeed: <i>boolean</i> {read-only, access after open}
Remarks	If true, the application can change the speed. If false, the application cannot change the speed. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	Speed Property

CapVoice Property

Syntax	CapVoice: <i>boolean</i> {read-only, access after open}
Remarks	If true, the application can change the voice. If false, the application cannot change the voice. This property is initialized by the open method.
Errors	A UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	Voice Property

UPOS Ver1.16 RCSD Specification

CapVolume Property

- Syntax** **CapVolume: *boolean* {read-only, access after open}**
- Remarks** If true, the application can change the volume. If false, the application cannot change the volume.
- This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
- See Also** **Volume** Property

Language Property

- Syntax** **Language: *string* {read-write, access after open-claim-enable}**
- Remarks** Indicates the language to speak. Valid values are one of the values listed in the **LanguageList** property.
- This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
- Some possible values of the exception’s **ErrorCode** property are:
- | <u>Value</u> | <u>Meaning</u> |
|--------------|---|
| E_ILLEGAL | An invalid value was specified. Or an unsupported language was specified. |
- See Also** **speak** Method, **speakImmediate** Method

LanguageList Property

- Syntax** **LanguageList: *string* {read-only, access after open}**
- Remarks** Contains the comma-delimited list of language that are supported by the device. The value representing the language is a value consisting of the language and country code defined in RFC 4664. For example, when the device supports US / English, Japan / Japanese, it will be as follows. "en-US, ja-JP"
- This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
- See Also** **Language** Property

OutputIDList Property

- Syntax** **OutputIDList : *string* {read-write, access after open-claim-enable}**
- Remarks** Comma-separated list of **OutputID** property values of audio being played by **Sspeak** method or **SspeakImmediate** method. This list indicates the capability how many and what kinds of utterance can be done by the targeted Speech Synthesis device
- This property is initialized by the **open** method. It will also be updated as the speech request increases or decreases.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
- See Also** **speak** Method, **speakImmediate** Method

UPOS Ver1.16 RCSD Specification

Pitch Property

- Syntax** Pitch: *int32* {read-write, access after open-claim-enable}
- Remarks** Holds the pitch at speech. Legal values range from 50% through 200%.
This property is initialized to 100% by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
Some possible values of the exception’s **ErrorCode** property are:
- | <u>Value</u> | <u>Meaning</u> |
|--------------|---------------------------------|
| E_ILLEGAL | An invalid value was specified. |
- See Also** speak Method, speakImmediate Method

Speed Property

- Syntax** Speed: *int32* {read-write, access after open-claim-enable}
- Remarks** Holds the speed at speech. Legal values range from 50% through 200%.
This property is initialized to 100% by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
Some possible values of the exception’s **ErrorCode** property are:
- | <u>Value</u> | <u>Meaning</u> |
|--------------|---------------------------------|
| E_ILLEGAL | An invalid value was specified. |
- See Also** speak Method, speakImmediate Method

Voice Property

- Syntax** Voice : *string* {read-write, access after open-claim-enable }
- Remarks** Indicates the voice tone to speak. Valid values are one of the values listed in the **VoiceList** property.
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.
Some possible values of the exception’s **ErrorCode** property are:
- | <u>Value</u> | <u>Meaning</u> |
|--------------|--|
| E_ILLEGAL | An invalid value was specified. Or an unsupported voice was specified. |
- See Also** speak Method, speakImmediate Method

UPOS Ver1.16 RCSD Specification

VoiceList Property

- Syntax** **VoiceList: *string* { read-only, access after open }**
- Remarks** A list of speech able voices ~~are~~is shown in a comma-separated list. For example, when the device supports male and female voice tones, it looks like the following.
"MALE_VOICE, FEMALE_VOICE"
(The content of the value depends on the device)

This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see "**Errors**" on page Intro-20.
- See Also** **Voice** Property

Volume Property

- Syntax** **Volume : *int32* {read-write, access after open-claim-enable}**
- Remarks** Holds the volume at speech. Legal values range from zero through 100.

This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this property is accessed. For further information, see "**Errors**" on page Intro-20.

Some possible values of the exception's **ErrorCode** property are:
- | <u>Value</u> | <u>Meaning</u> |
|--------------|---------------------------------|
| E_ILLEGAL | An invalid value was specified. |
- See Also** **speak** Method, **speakImmediate** Method

UPOS Ver1.16 RCSD Specification Methods (UML operations)

speak Method

Syntax `speak (text: string):
void {raises-exception, use after open-claim-enable}`

Parameter	Description
-----------	-------------

<i>Text</i>	Specify the text to speak.
-------------	----------------------------

Remarks ~~Device will utter the words specified by Text.~~ Device utters after converting the specified string into speech.

The utterance is executed according to the setting contents of **Language** property, **Volume** property, **Pitch** property, **Speed** property, but by inserting the following tag in the text, it is possible to change the utterance after the tag.

Tag	Description
<i>volume</i>	Specify the volume of the uttered voice. Valid values range from 1 to 100.
<i>pitch</i>	Specify the high or low of the uttered voice. Valid values range from 50 to 200.
<i>speed</i>	Specify the speed of the uttered voice. Valid values range from 50 to 200.
<i>pause</i>	Specify the time to pause in milliseconds.
<i>reset</i>	Delete the effect of volume, pitch, speed.

Content written in text is uttered with the following parameter settings.

Tag	Description	Value (decimal integer)	Default Value (decimal integer)
<i>volume</i>	Specify the volume of the uttered voice.	1 to 100	50
<i>pitch</i>	Specify the high or low of the uttered voice.	50 to 200	100
<i>speed</i>	Specify the speed of the uttered voice.	50 to 200	100
<i>pause</i>	Specify the time to pause in milliseconds.	1 to 50000	1
<i>reset</i>	Rest the effect of volume, pitch, speed to the default value.	-	-

[Goto Table 1-208](#)

UPOS Ver1.16 RCSD Specification

~~Tags without reset are specified in the form of "\\ tag = value \". For example, when specifying Text as follows, "Hello \\ pause = 1000 \\ \\ pitch = 150 \\ \\ It's nice weather today \\ \\ reset \". "Hello" speaks according to the original setting. Then wait for 1000 milliseconds. "Today" speaks Pitch at 150%. "Nice weather," I will speak according to the original settings.~~

If dialogue is " Hello. Today, it's nice weather."

Then if you would like to use the default setting of speed, volume, pitch for the "Hello". And would like to put a pose between "Hello" and "Today" 1000 milliseconds and would like to change the speaking pith of "Today" to 150 and increase the volume to 80. Then for the "It's nice weather" would like return to the default value by using the reset. It is described as follows

Hello.{pause=1000,pitch=150,volu=80}Today,{reset}It's nice weather.

Those utterance defined as follows.

Name	Data	Remarks	
Utterance written by text with the speak method parameter. Text will be spoken under the assigned parameter condition.	{#=f}XXXX{#=f}YYYY	#:Tag names It is volume, pitch, speed, pause and reset.	f:Tag values It is described in the Tag Value Table.

When this method is called by the application, device validate the method parameters, and if validation is successful buffer the request in program memory and deliver it to the device and process it. And device sets the unique integer identifier into the **OutputID** property. When device successfully complete a request an **OutputCompleteEvent** is enqueued for delivery to the application.

If the device does not support volume change etc., that tag will be ignored. This method is executed asynchronously. To end an utterance halfway, call the **stopCurrentSpeaking** method or the **stopSpeaking** method.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.
Some possible values of the exception's **ErrorCode** property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified. The language set in the Language property and the language specified by Text do not match.

See Also **Language** Property, **Volume** Property, **Pitch** Property, **OutputID** Property, **Speed** Property, **stopCurrentSpeaking** Method, **stopSpeaking** Method

[Goto Table 1-208](#)

UPOS Ver1.16 RCSD Specification

SpeakImmediate Method

Syntax `SpeakImmediate (text: string): void {raises-exception, use after open-claim-enable}`

Parameter	Description
-----------	-------------

<i>text</i>	Specify the text to speak.
-------------	----------------------------

Remarks The **speak** method acts to start speaking the words specified by text, while the **speakImmediate** method ends immediately previous **speak** method, and starts speaking the word specified by text asynchronously and immediately.

After executing the same processing as the **clearOutput** method, speak the wording specified by text.

Like this **speak** method, this method can also change a specific wording by inserting a tag. For details, refer to the description of **speak** method.

This method is executed asynchronously. To end an utterance halfway, call the **stopCurrentSpeaking** method or the **stopSpeaking** method.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20. Some possible values of the exception’s **ErrorCode** property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified. The language set in the Language property and the language specified by Text do not match.

See Also **Language** Property, **Volume** Property, **Pitch** Property, **Speed** Property, [speak Method](#), [stopCurrentSpeaking Method](#), [stopSpeaking Method](#)

[Goto Table 1-209](#)

stopCurrentSpeaking Method

Syntax `stopCurrentSpeaking (): void {raises-exception, use after open-claim-enable}`

Remarks ~~Stops the currently executed utterance.~~
The **speak** method and **speakImmediate** method start the speaking words specified by text and ends when **stopCurrentSpeaking** method is called. This method handles asynchronously.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.

Some possible values of the exception’s **ErrorCode** property are:

Value	Meaning
E_ILLEGAL	Speech is not running.

See Also [speak Method](#), [speakImmediate Method](#)

[Goto Table 1-210](#)

UPOS Ver1.16 RCSD Specification

stopSpeaking Method

Syntax **stopSpeaking** (*outputID* : *int32*):
 void {raises-exception, use after open, claim, enable}

<u>Parameter</u>	<u>Description</u>
<i>outputID</i>	Specify the value of the OutputID property you wish to terminate.

Remarks Stop and delete the utterance specified in OutputID.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.

Some possible values of the exception's **ErrorCode** property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

See Also **OutputID** Property, **speak** Method, **speakImmediate** Method

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

DirectIOEvent

<<event>> **upos::events::DirectIOEvent**

EventNumber : *int32* {read-only}
Data : *int32* {read-write}
Obj : *object* {read-write}

Description Provides Service information directly to the application. This event provides a means for a vendor-specific Sound Player Service to provide events to the application that are not otherwise supported by the device control.

Attributes This event contains the following attributes:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>EventNumber</i>	<i>int32</i>	Event number whose specific values are assigned by the Service.
<i>Data</i>	<i>int32</i>	Additional numeric data. Specific values vary by the <i>EventNumber</i> and the Service. This attribute is settable.
<i>Obj</i>	<i>object</i>	Additional data whose usage varies by the <i>EventNumber</i> and the Service. This attribute is settable.

Remarks This event is to be used only for those types of vendor specific functions that are not otherwise described.

Use of this event may restrict the application program programform being used with other vendor's devices which may not have any knowledge of the Service's need for this event.

See Also "Events" on page Intro-19, **directIO** method

[Goto Table 1-211](#)

UPOS Ver1.16 RCSD Specification

ErrorEvent

```
<<event>> upos::events:: ErrorEvent
  ErrorCode           : int32{read-write}
  ErrorCodeExtended   : int32{read-write}
  ErrorLocus          : int32{read-write}
  ErrorResponse       : int32{read-write}
```

Description Notifies the application that a Speech Synthesis Device error has been detected and suitable response by the application is necessary to process the error condition.

Attributes This event contains the following attributes:

Attributes	Type	Description
<i>ErrorCode</i>	<i>int32</i>	Error code causing the error event. See a list of Error Codes on page 20.
<i>ErrorCodeExtended</i>	<i>int32</i>	Extended Error code causing the error event. If <i>ErrorCode</i> is E_EXTENDED, then see values below. Otherwise, it may contain a Service-specific value.
<i>ErrorLocus</i>	<i>int32</i>	Location of the error. If EL_OUTPUT is specified. An error occurred during asynchronous action.
<i>ErrorResponse</i>	<i>int32</i>	Error response, whose default value may be overwritten by the application (i.e., this attribute is settable). See values below.

The *ErrorLocus* attribute has one of the following values:

Value	Meaning
EL_OUTPUT	Error occurred while processing asynchronous output.

The application's error event handler can set the *ErrorResponse* attribute to one of the following values:

Value	Meaning
ER_RETRY	Retry sending the data. The error state is exited. Typically, valid for asynchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited. This is the default response when the locus is EL_OUTPUT.
ER_CLEAR	Valid for loci: EL_OUTPUT. Clear all buffered input or output data (including all asynchronous output). The error state is exited.

Remarks This event is enqueued when an error is detected and the Device's **State** transitions into the error state.

See Also "Error Handling" on page Intro-23, "Device Output Models" on page Intro-25.

[Goto Table 1-212](#)

UPOS Ver1.16 RCSD Specification

OutputCompleteEvent

<<event>> **upos::events::OutputCompleteEvent**
OutputID : *int32*{read-only}

Description Notify the application that the queued output request associated with the *outputID* property has completed successfully.

Attributes This event contains the following attributes:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>OutputID</i>	<i>int32</i>	The ID number of the asynchronous output request that is complete.

Remarks This event is enqueued after the request's data has been both sent, and the Service has confirmation that it was processed by the device successfully.

See Also "Device Output Models" on page Intro-25

[Goto Table 1-213](#)

StatusUpdateEvent

<<event>> **upos::events::StatusUpdateEvent**
Status : *int32* {read-only}

Description *Notifies the application that there is an operation status change or a status of the Speech Synthesis device.*

Attributes This event contains the following attribute:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>Status</i>	<i>int32</i>	Indicates a change of operation status of sound player device

Note that Release 1.3 added Power State Reporting with additional Power reporting StatusUpdateEvent values.

The Update Firmware capability added additional *Status* values for communicating the status/progress of an asynchronous update firmware process. See "StatusUpdateEvent" description on page 1-34.

<u>Value</u>	<u>Meaning</u>
SPCH_SUE_START_SPEAK	It will be notified when speech synthesis starts.
SPCH_SUE_STOP_SPEAK	It will be notified when speech synthesis stops.

Remarks Enqueued when the Speech Synthesis Device detects a power state change or a status change.

See Also "Events" on page Intro-19.

[Goto Table 1-214](#)

Gesture Control

This Chapter defines the Gesture Control device category.

Summary

Properties (UML attributes)

<i>Common</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
AutoDisable:	<i>boolean</i>	{read-write}	---	Not supported open
CapCompareFirmwareVersion:	<i>boolean</i>	{read-only}	1.16	open
CapPowerReporting:	<i>int32</i>	{read-only}	1.16	open
CapStatisticsReporting:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateFirmware:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateStatistics:	<i>boolean</i>	{read-only}	1.16	open
CheckHealthText:	<i>string</i>	{read-only}	1.16	open
Claimed:	<i>boolean</i>	{read-only}	1.16	open
DataCount:	<i>int32</i>	{read-only}	---	Not supported open
DataEventEnabled:	<i>boolean</i>	{read-write}	---	Not supported open
DeviceEnabled:	<i>boolean</i>	{read-write}	1.16	open & claim
FreezeEvents:	<i>boolean</i>	{read-write}	1.16	open
OutputID:	<i>int32</i>	{read-only}	1.16	open
PowerNotify:	<i>int32</i>	{read-write}	1.16	open
PowerState:	<i>int32</i>	{read-only}	1.16	open
State:	<i>int32</i>	{read-only}	1.16	--
DeviceControlDescription:	<i>string</i>	{read-only}	1.16	--
DeviceControlVersion:	<i>int32</i>	{read-only}	1.16	--
DeviceServiceDescription:	<i>string</i>	{read-only}	1.16	open
DeviceServiceVersion:	<i>int32</i>	{read-only}	1.16	open
PhysicalDeviceDescription:	<i>string</i>	{read-only}	1.16	open
PhysicalDeviceName:	<i>string</i>	{read-only}	1.16	open

[Goto Table 1-215](#)

UPOS Ver1.16 RCSD Specification

Properties (Continued)

<i>Specific</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
CapAssociatedHardTotalsDevice:	<i>string</i>	{read-only}	1.16	open
CapMotion:	<i>boolean</i>	{read-only}	1.16	open
CapMotionCreation:	<i>boolean</i>	{read-only}	1.16	open
CapPose:	<i>boolean</i>	{read-only}	1.16	open
CapPoseCreation:	<i>boolean</i>	{read-only}	1.16	open
CapStorage:	<i>int32</i>	{read-only}	1.16	open
AutoMode:	<i>string</i>	{read-write}	1.16	open, claim & enable
AutoModeList:	<i>string</i>	{read-only}	1.16	open
JointList:	<i>string</i>	{read-only}	1.16	open
MotionList:	<i>string</i>	{read-only}	1.16	open
PoseCreationMode:	<i>boolean</i>	{read-write}	1.16	open, claim & enable
PoseList:	<i>string</i>	{read-only}	1.16	open
Storage:	<i>int32</i>	{read-write}	1.16	open, claim & enable

Methods (UML operations)

Common

<i>Name</i>	<i>Version</i>
open (logicalDeviceName: <i>string</i>): void {raises-exception}	1.16
close (): void {raises-exception, use after open}	1.16
claim (timeout: <i>int32</i>): void {raises-exception, use after open}	1.16
release (): void {raises-exception, use after open, claim}	1.16
checkHealth (level: <i>int32</i>): void {raises-exception, use after open, enable}	1.16
clearInput (): void { }	1.16 Not supported
clearInputProperties (): void { }	1.16 Not supported
clearOutput (): void { }	1.16 Not supported
compareFirmwareVersion (firmwareFileName: <i>string</i> , out result: <i>int32</i>): void {raises-exception, use after open, enable}	1.16
directIO (command: <i>int32</i> , inout data: <i>int32</i> , inout obj: <i>object</i>): void {raises-exception, use after open}	1.16
resetStatistics (statisticsBuffer: <i>string</i>): void {raises-exception, use after open, enable}	1.16
retrieveStatistics (inout statisticsBuffer: <i>string</i>): void {raises-exception, use after open, enable}	1.16
updateFirmware (firmwareFileName: <i>string</i>): void {raises-exception, use after open, enable}	1.16
updateStatistics (statisticsBuffer: <i>string</i>): void {raises-exception, use after open, enable}	1.16

[Goto Table 1-217](#)
[Goto Table 1-297](#)

UPOS Ver1.16 RCSD Specification

Methods (UML operations)(continued)

Specific

<i>Name</i>	<i>Version</i>
createMotion (fileName: <i>string</i> , poseList: <i>string</i>): void { raises-exception, use after open, claim, enable }	1.16
createPose (fileName: <i>string</i> , time: <i>int32</i>): void { raises-exception, use after open, claim, enable }	1.16
getPosition (jointID: <i>string</i> , out position: <i>int32</i> by reference): void { raises-exception, use after open, claim, enable }	1.16
setPosition (positionList: <i>string</i> , time: <i>int32</i> , absolute: <i>boolean</i>): void { raises-exception, use after open, claim, enable }	1.16
setSpeed (speedList: <i>string</i> , time: <i>int32</i>): void { raises-exception, use after open, claim, enable }	1.16
startMotion (fileName: <i>string</i>): void { raises-exception, use after open, claim, enable }	1.16
startPose (fileName: <i>string</i>): void { raises-exception, use after open, claim, enable }	1.16
stopControl (outputID: <i>int32</i>): void { raises-exception, use after open, claim, enable }	1.16

[Goto Table 1-219](#)

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

<i>Name</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>
upos::events::DataEvent		<i>Not supported</i>	
upos::events::DirectIOEvent			1.16
EventNumber:	<i>int32</i>	{read-only}	
Data:	<i>int32</i>	{read-write}	
Obj:	<i>object</i>	{read-write}	
upos::events::ErrorEvent			1.16
ErrorCode:	<i>int32</i>	{read-only}	
ErrorCodeExtended:	<i>int32</i>	{read-only}	
ErrorLocus:	<i>int32</i>	{read-only}	
ErrorResponse:	<i>int32</i>	{read-write}	
upos::events::OutputCompleteEvent			1.16
OutputID:	<i>int32</i>	{read-only}	
upos::events::StatusUpdateEvent			1.16
Status:	<i>int32</i>	{read-only}	
upos::events::TransitionEvent		<i>Not supported</i>	

[Goto Table 1-219](#)

UPOS Ver1.16 RCSD Specification

General Information

The Gesture Control device programmatic name is "Gesture Control".

Capabilities

The Gesture Control device has the following capability:

- It controls the behavior of various joint components and parts.
- The operation is automatically controlled by interlocking various joints and other devices.
- Register and play the defined pose and motion.

Gesture Control Class Diagram

The following diagram shows the relationships between the Gesture Control classes.

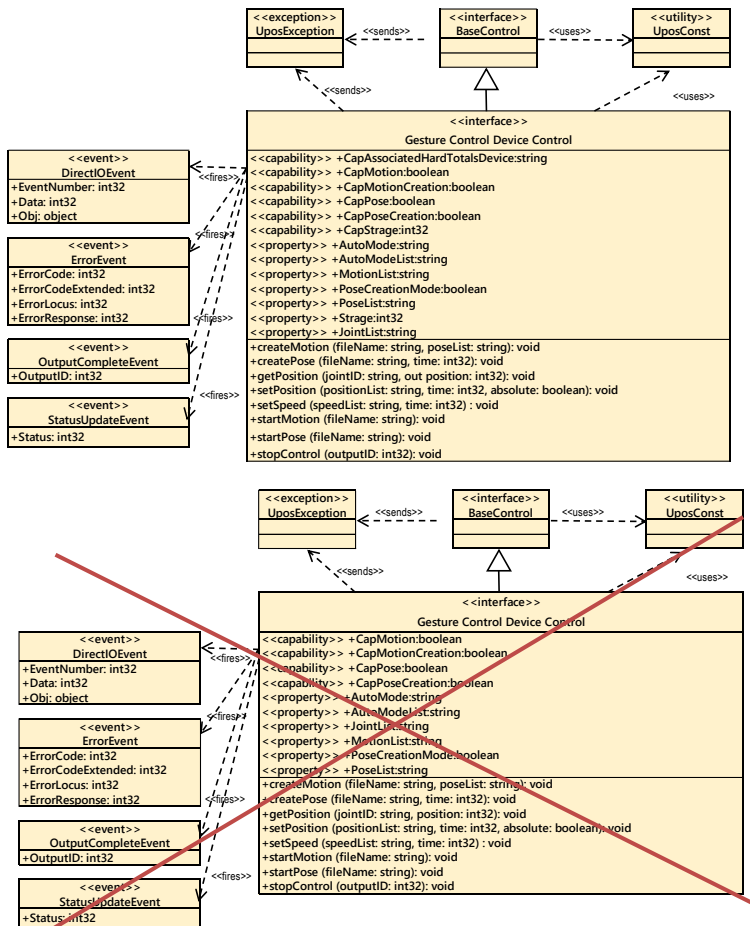


Fig. Chap. 45-1 Gesture Control Class Diagram

[Goto Table2-7](#)

UPOS Ver1.16 RCSD Specification

Model

The Gesture Control follows the general device behavior model for asynchronous output devices:

- The application calls a **setPosition**, **setSpeed**, **startPose**, **startMotion** method to start output. The Device validates the method parameters and produces an error condition immediately if necessary. If the validation is successful, the Device does the following:
 - **+** Buffers the request in program memory, for delivery to the Physical Device as soon as the Physical Device can receive and process it.
 - **⌘** Sets the **OutputID** property to a unique integer identifier for this request.
 - **⌘** Returns as soon as possible.
- When the Device successfully completes a request, an **OutputCompleteEvent** is enqueued for delivery to the application. A property of this event contains the outputID of the completed request. The application should compare the returned **OutputCompleteEvent** property OutputID value with the OutputID value set by the asynchronous process method call used to send the data, in order to track what data has been successfully sent to the device.
- If an error occurs while processing a request, an **ErrorEvent** is enqueued which will be delivered to the application after the events already enqueued, including **OutputCompleteEvent**. No further asynchronous output will occur until the event has been delivered to the application. If the response is ER_CLEAR, then outstanding asynchronous output is cleared. If the response is ER_RETRY, then output is retried; note that if several outputs were simultaneously in progress at the time that the error was detected, then the Service may need to retry all of these outputs.
- Asynchronous output is always performed on a first-in first-out basis.
- If the request is terminated before completion, due to reasons such as the application calling the **clearOutput** method, then no **OutputCompleteEvent** is delivered.
- Application can also delete the output individually by calling the **stopControl** method. Also, in this case **OutputCompleteEvent** will not be notified.
- The application will be informed about any status change with a **StatusUpdateEvent**, also all corresponding status properties will be updated before event delivery.
- [Goto Table 1-220](#)

Automatic control

Automatic control of a joint means to automatically control a joint on the device side, such as tracking according to the movement of a person's face, in cooperation with a camera or the like connected to the device.

The automatic control function is device dependent. For possible automatic control, it is enabled by confirming with the **AutoModeList** property and setting a value in the **AutoMode** property.

UPOS Ver1.16 RCSD Specification

Pose / Motion

Pose refers to setting the position of one or more defined joints.

For example, it is an action that lifts a hand.

To execute a pose, specify the pose file name by the **startPose** method or the pose name defined in the device.

Create the pose file with the **createPose** method described later. Pose defined in the device will be checked in the value of **PoseList** property.

To execute motion, specify the motion file name or the motion name defined in the device with the **startMotion** method.

Motion files are created by the **createMotion** method to be described later. Motion defined in the device can be checked with the value of **MotionList** property.

To create a pose file, first set the **PoseCreationMode** property to TRUE and enable the pose registration function. When pose registration function is enabled, each joint is set to the default position. At this time, if the automatic control mode is enabled, the automatic control mode is temporarily invalidated.

Then, an application can ~~then~~ create a pose file by setting the value ~~you want to be~~ defined as a pose with the **setPosition** method and calling the **createPose** method.

A motion file can be created ~~and recorded~~ by specifying the pose defined ~~by~~ in the created pose file or ~~the pose defined in the device~~ and creating it as a series of ~~continuously changing actions~~ and calling the **createMotion** method.

Since the created ~~pose~~ and motion files are recorded in the area ~~managed by~~ may store in either the “**Hard Totals**” ~~service, the application must also support “Hard Totals” service~~ devices or the host file system, or both, and the **CapStorage** property will show the device’s data file storage location capability.

If device supports either of both Hard Totals devices and the host file system, the application should set the **Storage** property accordingly to tell where to write the data file.

If device needs to be able to write the pose and motion files to a Hard Totals device, the **CapAssociatedHardTotalsDevice** property holds the open name of the associated Hard Totals device.

[Goto Table 1-221](#)

Device Sharing

The Gesture Control **device** is an exclusive-use device, as follows:

- The application must claim the device before enabling it.
- The application must claim and enable the device before accessing some properties or calling methods that update the device.
- See the “Summary” table for precise usage prerequisites.

UPOS Ver1.16 RCSD Specification

Properties (UML attributes)

AutoMode Property

Syntax	AutoMode: <i>string</i> {read-write, access after open-claim-enable}				
Remarks	<p>Indicates automatic control mode ID. Valid values are the empty string "" or one of the AutoModeList properties listed.</p> <p>If you set one of the properties described in the AutoModeList property is set for this property, the automatic control mode will be enabled in the set mode.</p> <p>Setting the empty character "" disables the automatic control mode.</p> <p>This property is initialized to the empty string "" by the open method.</p>				
Errors	<p>A UposException may be thrown when this method is invoked. For further information, see “Errors” on page Intro-20.</p> <p>Some possible values of the exception’s <i>ErrorCode</i> property are:</p> <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified.</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	E_ILLEGAL	An invalid value was specified.
<u>Value</u>	<u>Meaning</u>				
E_ILLEGAL	An invalid value was specified.				
See Also	AutoModeList Property Goto Table 1-222				

AutoModeList Property

Syntax	AutoModeList: <i>string</i> {read-only, access after open}
Remarks	<p>Comma-separated list of joint automatic control IDs supported by the device.</p> <p>For example, in conjunction with the camera, if the mode of tracking the face of a person by moving only the joint of Joint01, and the mode of tracking by moving all joints are supported as follows, this is “FaceTrack_Joint01”.</p> <p>“FaceTrack_Joint01,FaceTrack_ALL”</p> <p>Another example, in conjunction with the camera, if the mode of tracking the face of a person by moving all joints are supported, this is “FaceTrack_ALL”.</p> <p>(Content and order are dependent on the device.)</p> <p>This property is initialized by the open method.</p>
Errors	<p>A UposException may be thrown when this method is invoked. For further information, see “Errors” on page Intro-20.</p>
See Also	AutoMode Property. Goto Table 1-223

CapAssociatedHardTotalsDevice Property

Syntax	CapAssociatedHardTotalsDevice : <i>string</i> {read-only, access after open}
Remarks	<p>Holds the open name of the associated Hard Totals device if the device is able to write to such devices which is the case if CapStorage is either GCTL_CST_ALL or GCTL_CST_HARDTOTALS_ONLY. If CapStorage is GCTL_CST_HOST_ONLY this property value must be the empty string.</p>
Errors	<p>UposException may be thrown when this property is accessed. For further information, see “Errors” on page Intro-20.</p>
See Also	CapStorage Property Goto Table 1-224

UPOS Ver1.16 RCSD Specification

CapMotion Property

Syntax	CapMotion: <i>boolean</i> {read-only, access after open}
Remarks	If true, the device supports pose making the motion function. Otherwise, it is false. If false, the device does not support pose function. If When this property is false, change of PoseCreationMode property, startPose method, createPose method is not available. startMotion method, createMotion method is not available. This property is initialized by the open method.
Errors	A UposException may be thrown when this method is invoked. For further information, see “ Errors ” on page Intro-20.
See Also	startMotion Method, createMotion Method. Goto Table 1-225

CapMotionCreation Property

Syntax	CapMotionCreation: <i>boolean</i> {read-only, access after open}
Remarks	If true, the device supports motion registration function. If false, the device does not support motion registration function. If this property is FALSE, the createMotion method is not available. This property is initialized by the open method.
Errors	A UposException may be thrown when this method is invoked. For further information, see “ Errors ” on page Intro-20.
See Also	createMotion Method.

CapPose Property

Syntax	CapPose: <i>boolean</i> {read-only, access after open}
Remarks	If true, the device supports pose function. Otherwise, it is false. If false, the device does not support pose function. If When this property is FALSE, change of PoseCreationMode property value cannot be changed, in addition, startPose method, and createPose method is are not available. This property is initialized by the open method.
Errors	A UposException may be thrown when this method is invoked. For further information, see “ Errors ” on page Intro-20.
See Also	PoseCreationMode Property, startPose Method, createPose Method. Goto Table 1-226

CapPoseCreation Property

Syntax	CapPoseCreation: <i>boolean</i> {read-only, access after open}
Remarks	If true, the device supports pose registration function. If false, the device does not support pose registration function. If When this property is FALSE, you cannot use the createPose method that can to change the PoseCreationMode property is not available. This property is initialized by the open method.
Errors	A UposException may be thrown when this method is invoked. For further information, see “ Errors ” on page Intro-20.
See Also	PoseCreationMode Property, createPose Method. Goto Table 1-227

UPOS Ver1.16 RCSD Specification

CapStorage Property

Syntax CapStorage: *int32* {read-only, access after open}

Remarks This is an enumeration and announces where the device is able to write the recorded motion and/or pose data file to. It holds one of the following values.

<u>Value</u>	<u>Meaning</u>
GCTL_CST_HARDTOTALS_ONLY	Only an associate Hard Totals device is supported.
GCTL_CST_HOST_ONLY	Only the host's file system is supported.
GCTL_CST_ALL	Both, the associated Hard Totals device and the host's file system is supported.

This property is initialized by the **open** method.

If a Hard Totals device is supported the **Storage** the property value should be GCTL_CST_HARDTOTALS_ONLY or GCTL_CST_ALL, and the property **CapAssociatedHardTotalsDevice** holds the open name of the associated **Hard Totals** device.

Errors UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20.

See Also **Storage** Property, **CapAssociatedHardTotalsDevice** Property

[Goto Table 1-228](#)

UPOS Ver1.16 RCSD Specification

JointList Property

Syntax **JointList:** *string* {read-only, access after open}

Remarks Comma-separated list of joint information supported by the device.

Each piece of joint information consists of the following information and is shown in the following order, separated by a colon (":").

<u>Parameter</u>	<u>Description</u>
<i>JointID</i>	Indicates a unique ID in the service that identifies the joint. Position range availability: If 0, the joint does not have the position range, 1 holds the position range. For example, the arm joint has a range of rotation width, but the wheel for movement does not have the range of movement amount. If position range is 0, the Joint does not have the position range. If position range is 1, the joint holds the position range. For example, for a device that supports pitch, roll, and yaw joints and a device that supports rotation by wheel and joint that can move forward and backward, it is as follows. — For example, arm joint has a range of rotation width but wheel for movement does not have the range of movement amount. If there is a device with joints that supports pitch, roll, yaw and wheels that supports rotating and moving back and forth. In this case they are indicated as follows: "Joint01_Pitch:1, Joint01_Roll:1, Joint01_Yaw:1, Wheel_Turn:0, Wheel_Move:0"

This property is initialized by the **open** method.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20. [Goto Table 1-229](#)

MotionList Property

Syntax **MotionList:** *string* {read-only, access after open}

Remarks Comma-separated list of motion IDs defined on the device.

For example, "bowing, welcoming, clapping..."

This property is initialized by the **open** method.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20. [Goto Table 1-230](#)

UPOS Ver1.16 RCSD Specification

PoseCreationMode Property

Syntax **PoseCreationMode:** *boolean* {read-write, access after open-claim-enable}

Remarks If true, pose registration function is enabled.
If false, pose registration function is invalid.
When this property is set to true, pose registration function is enabled. When false is set, the pose registration function is disabled.
This property is initialized to false when you first enable the device after calling the **open** method.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.
Some possible values of the exception’s *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

See Also **CapPose** Property, **CapPoseCreation** Property.

PoseList Property

Syntax **PoseList:** *string* {read-only, access after open}

Remarks A comma-separated list of pose IDs defined on the device.
[For example, “surprise, bow, think,...”](#)
This property is initialized by the **open** method.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.

[Goto Table 1-231](#)

UPOS Ver1.16 RCSD Specification

Storage Property

Syntax Storage: *int32* {read-write, access after open-claim-enable}

Remarks This is an enumeration and defines where the device writes the recorded motion and/or pose data file to. Should be set before an appropriate method call.
It holds one of the following values.

<u>Value</u>	<u>Meaning</u>
GCTL_ST_HARDTOTALS	The motion and/or pose data file is written to the associated Hard Totals device. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device.
GCTL_ST_HOST	The motion and/or pose data file is written to the host's file system.
GCTL_ST_HOST_HARDTOTALS	The motion and/or pose data file is written to the associated Hard Totals device and host's file system. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device.

This property is initialized by the **open** method according to the value hold by **CapStorage**. If **CapStorage** has the value GCTL_CST_ALL, it is initialized to GCTL_ST_HOST_HARDTOTALS.

Errors UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified, or recording is ongoing.

See Also **CapStorage** Property, **CapAssociatedHardTotalsDevice** Property

[Goto Table 1-232](#)

UPOS Ver1.16 RCSD Specification

Table of Gesture Control Device Listed Items in Property

Property Name	Item ID, File Name, Name	Parameter
AutoModeList	Face Track	Joint01 Joint_ALL
	Chase	Joint01, Wheel01, Wheel02 Joint_All, Wheel_ALL,
MotionList	Bowing, Welcoming, Clapping, Farewelling01, Farewelling02, Greeting01, Greeting02 ,	
PoseList	Surprise, Bow01, Bow02, Think01, Think02 Doubt01, Doubt02	
JointList	Joint	Pitch Roll Yaw
	Wheel	Turn Move Back Move Forth

[Goto Table 1-233](#)

Methods (UML operations)

createMotion Method

Syntax createMotion (fileName: *string*, poseList: *string*):
void {raises-exception, use after open-claim-enable}

Parameter	Description
<i>fileName</i>	Specify the motion file name to be recorded as motion.
<i>poseList</i>	Specify the comma-separated list of pose information to be registered.

Remarks ~~Specify the registered pose and record it in the motion file.~~
A motion file can be created and recorded by specifying the pose defined in the created pose file or the pose defined in the device and creating it as a series of continuously changing actions.
The place where the motion file is recorded is the area ~~managed by the "hard totals" device~~ value of the **Storage** property.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.
Some possible values of the exception's *ErrorCode* property are:

Value	Meaning
E_ILLEGAL	fileName is too long or contains unusable characters.
E_EXISTS	fileName already exists.

[Goto Table 1-234](#)

createPose Method

Syntax createPose (fileName: *string*, time: *int32*):
void {raises-exception, use after open-claim-enable}

Parameter	Description
<i>fileName</i>	Specify the pose file name to record the pose.
<i>time</i>	Specify the time to reach the pose position.

Remarks Record the position of each joint in the pose file.
Before calling this method, ~~you~~ it needs to set the **PoseCreationMode** property to TRUE and to make enableing pose registration mode.
The place where the motion file is recorded is the area ~~managed by the "hard totals" device~~ value of the **Storage** property.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.
Some possible values of the exception's *ErrorCode* property are:

Value	Meaning
E_ILLEGAL	FileName is too long or contains unusable characters. Or PoseCreationMode is FALSE.
E_EXISTS	FileName already exists.

See Also **PoseCreationMode** Property.

[Goto Table 1-235](#)

UPOS Ver1.16 RCSD Specification

getPosition Method

Syntax getPosition (jointID: *string*, **out** position: *int32* ~~*by-reference*~~):
 void {raises-exception, use after open-claim-enable}

<u>Parameter</u>	<u>Description</u>
<i>jointID</i>	Specify the one of the joint ID values that are listed in the JointList property. Specify one of the values listed in the JointList property. However, it must be an ID whose position range exists or not. And specified JointList property should be the position range present one.
<i>position</i>	The position of the joint specified by JointID is stored. Store the specified value as the position associated with jointID.

Remarks ~~It acquires the position specified by jointID and stores it in position.~~
It acquires the position specified by jointID and stores it in position.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.
Some possible values of the exception’s *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

See Also **JointList** Property.

[Goto Table 1-236](#)

UPOS Ver1.16 RCSD Specification
setPosition Method

Syntax **setPosition (positionList: string, time: int32, absolute: boolean);**
 void {raises-exception, use after open-claim-enable}

Parameter	Description
<i>positionList</i>	Specify the position information in a comma-separated list.
<i>time</i>	Specify the time of device to control completion in seconds. If this value is too small, it will be changed to an appropriate value depending on the service.
<i>absolute</i>	If true, the specified position indicates the absolute value. If false, the specified position indicates relative value.

Each position information specified in the positionList consists of the following information and is shown in the following order separated by a colon (":").

Parameter	Description
<i>jointID</i>	Specify the joint ID. Specify one of the values listed in the JointList property. However, it must be an ID whose position range is present exists or not .
<i>position</i>	Specify the position to be set. Valid values range from -100 to 100. 100 represents the limit value in the positive direction of the target joint, and -100 represents the limit value in the negative direction. If absolute is a relative value (false) and the value specified here exceeds the limit value, it will be changed to an appropriate value by the service

For example, to move Yaw of Joint01 up to the limit of the positive direction and move Pitch of Joint02 to the middle, specify as follows.
 "Joint01_Yaw:100,Joint02:Pitch:0"

Remarks The joint position is set with the contents specified in PositionList and device control is started so that device control is completed at the time specified by Time.

Joints that can be specified with this method are only those that have a position range.

Check the **JointList** property for the presence or absence of the position range.

This method is executed asynchronously. To terminate the operation prematurely, call the **stopControl** method.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.

Some possible values of the exception's *ErrorCode* property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified.

See Also **JointList** Property, **stopControl** Method.

[Goto Table 1-237](#)

UPOS Ver1.16 RCSD Specification
setSpeed Method

Syntax **setSpeed (speedList: string, time: int32):**
 void {raises-exception, use after open-claim-enable}

Parameter	Description
<i>speedList</i>	Specify speed information in a comma-separated list.
<i>time</i>	Specify the time to device control in seconds. If you specify the value of FOREVER(-1) is specified , it will continue to operate until you call the stopControl method.

Each speed information specified in the SpeedList consists of the following information, and it is shown in the following order separated by a colon (":").

Parameter	Description
<i>jointID</i>	Specify the joint ID. Specify one of the values listed in the JointList property.
<i>speed</i>	Specify the speed to set. Valid values range from -100 to 100. 100 represents the maximum speed in the positive direction of the target joint, and -100 represents the maximum speed in the negative direction.

For example, to move Wheel's X at the maximum speed in the positive direction and Y at the Wheel at half the speed in the negative direction, specify as follows.

"Wheel_X:100, Wheel_Y:-50"

Remarks It sets the speed of the joint with the contents specified by speedList and performs **device** control for the time specified by time.

This method is executed asynchronously. To terminate the operation prematurely, call the **stopControl** method.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.

Some possible values of the exception's *ErrorCode* property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified.

See Also **JointList** Property, **stopControl** Method.

[Goto Table 1-238](#)

UPOS Ver1.16 RCSD Specification
startMotion Method

Syntax **startMotion (fileName: string):**
 void {raises-exception, use after open-claim-enable}

Parameter	Description
<i>fileName</i>	Specify the name of the motion file to start. Or one of the motion ID lists listed in the MotionList property. Prior to start this method, need to specify the name of the motion file or the motion ID value that is listed in the MotionList property.

Remarks Start the motion defined by fileName or motion defined by the device.
~~Motion files need to be placed in the area managed by "hard total" service. This method is executed asynchronously. To terminate motion control prematurely, call the stopControl method.~~
This method is executed asynchronously and when the device successfully completes a request, an **OutputCompleteEvent** is enqueued and a property of corresponding event's OutputID is placed into the **OutputID** property. The application should compare the returned **OutputCompleteEvent** property outputID value set by this method to track what data has been sent to device.
Motion files are placed in the area as the value of **Storage** property.
To terminate motion control prematurely, call the **stopControl** method.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.
Some possible values of the exception's *ErrorCode* property are:

Value	Meaning
E_ILLEGAL	An invalid value was specified.
E_NOEXIST	File does not exist.

See Also **MotionList** Property.

[Goto Table 1-239](#)

UPOS Ver1.16 RCSD Specification

startPose Method

Syntax **startPose (fileName: string):**
 void {raises-exception, use after open-claim-enable}

<u>Parameter</u>	<u>Description</u>
<i>fileName</i>	Specify the name of the pause pose file to start. Or one of the pose ID lists listed in the PoseList property.

Remarks ~~Begin pause~~ Start the pose defined by the ~~pose~~pose file or device specified by *fileName*.
 This method is executed asynchronously and when the device successfully completes a request, an **OutputCompleteEvent** is enqueued and a property of corresponding event's OutputID is placed into the **OutputID** property. The application should compare the returned **OutputCompleteEvent** property **OutputID** value set by this method to track what data has been sent to device.
~~Pose files must be placed in the area managed by "hard total" service.~~
 Pose files are placed in the area as the values of **Storage** property.
 To terminate pause control prematurely, call the **stopControl** method.

Errors A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20.
 Some possible values of the exception's *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.
E_NOEXISTS	File does not exist.

See Also **PoseList** Property, **stopControl** Method. [Goto Table 1-240](#)

stopControl Method

Syntax **stopControl (outputID: int32):**
 void {raises-exception, use after open-claim-enable}

<u>Parameter</u>	<u>Description</u>
<i>outputID</i>	Specify the value of the OutputID property to be you wish to terminated.

Remarks Stop the control specified for *outputID*. When device successfully complete the request, and **OutputCompleteEvent** is enqueued. A property of this event contains the *outputID* of the completed request. The application should compare the returned **OutputCompleteEvent** property OutputID value with OutputID value set by this method.

Errors A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20.
 Some possible values of the exception's *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

See Also **setPosition** Method, **setSpeed** Method, **startPose** Method, **startMotion** Method.

[Goto Table 1-241](#)

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

DirectIOEvent

<<event>> **upos::events::DirectIOEvent**

EventNumber : *int32* {read-only}
Data : *int32* {read-write}
Obj : *object* {read-write}

Description Provides Service information directly to the application. This event provides a means for a vendor-specific Sound Player Service to provide events to the application that are not otherwise supported by the device control.

Attributes This event contains the following attributes:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>EventNumber</i>	<i>int32</i>	Event number whose specific values are assigned by the Service.
<i>Data</i>	<i>int32</i>	Additional numeric data. Specific values vary by the <i>EventNumber</i> and the Service. This attribute is settable.
<i>Obj</i>	<i>object</i>	Additional data whose usage varies by the <i>EventNumber</i> and the Service. This attribute is settable.

Remarks This event is to be used only for those types of vendor specific functions that are not otherwise described.

Use of this event may restrict the application program programform being used with other vendor's devices which may not have any knowledge of the Service's need for this event.

See Also "Events" on page Intro-19, **directIO** method

[Goto Table 1-242](#)

UPOS Ver1.16 RCSD Specification

ErrorEvent

```
<<event>> upos::events:: ErrorEvent
  ErrorCode      : int32{read-write}
  ErrorCodeExtended : int32{read-write}
  ErrorLocus     : int32{read-write}
  ErrorResponse  : int32{read-write}
```

Description Notifies the application that a Gesture Control Device error has been detected and suitable response by the application is necessary to process the error condition.

Attributes This event contains the following attributes:

Attributes	Type	Description
<i>ErrorCode</i>	<i>int32</i>	Error code causing the error event. See a list of Error Codes on page 20.
<i>ErrorCodeExtended</i>	<i>int32</i>	Extended Error code causing the error event. If <i>ErrorCode</i> is E_EXTENDED, then see values below. Otherwise, it may contain a Service-specific value.
<i>ErrorLocus</i>	<i>int32</i>	Location of the error. If EL_OUTPUT is specified. An error occurred during asynchronous action.
<i>ErrorResponse</i>	<i>int32</i>	Error response, whose default value may be overridden by the application (i.e., this attribute is settable). See values below.

If *ErrorCode* is E_EXTENDED, then *ErrorCodeExtended* has one of the following values:

Value	Meaning
EGCTL_NOROOM	There is not enough room for the targeted data file storage area.

The *ErrorLocus* attribute has one of the following values:

Value	Meaning
EL_OUTPUT	Error occurred while processing asynchronous output.

The application's error event handler can set the *ErrorResponse* attribute to one of the following values:

Value	Meaning
ER_RETRY	Retry sending the data. The error state is exited. Typically, valid for asynchronous output devices when the locus is EL_OUTPUT, in which case the asynchronous output is re-tried, and the error state is exited. This is the default response when the locus is EL_OUTPUT.
ER_CLEAR	Valid for all loci: EL_OUTPUT. Clear all buffered input or output data (including all asynchronous output). The error state is exited.

Remarks This event is enqueued when an error is detected and the Device's **State** transitions into the error state.

See Also "Error Handling" on page Intro-23, "Device Output Models" on page Intro-25.

[Goto Table 1-243](#)

UPOS Ver1.16 RCSD Specification

OutputCompleteEvent

<<event>> **upos::events::OutputCompleteEvent**
OutputID: int32{read-only}

Description Notify the application that the queued output request associated with the *outputID* property has completed successfully.

Attributes This event contains the following attributes:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>OutputID</i>	<i>int32</i>	The ID number of the asynchronous output request that is complete.

Remarks This event is enqueued after the request's data has been both sent, and the Service has confirmation that it was processed by the device successfully.

See Also "Device Output Models" on page Intro-25

[Goto Table 1-244](#)

StatusUpdateEvent

<<event>> **upos::events::StatusUpdateEvent**
Status : int32 {read-only}

Description *Notifies the application that there is an operation status change or a status of the Gesture Control device.*

Attributes This event contains the following attribute:

<u>Attributes</u>	<u>Type</u>	<u>Description</u>
<i>Status</i>	<i>int32</i>	Indicates a change of operation status of sound player device

Note that Release 1.3 added Power State Reporting with additional Power reporting StatusUpdateEvent values.

The Update Firmware capability added additional *Status* values for communicating the status/progress of an asynchronous update firmware process. See "StatusUpdateEvent" description on page 1-34.

<u>Value</u>	<u>Meaning</u>
GCTL_SUE_START_MOTION	It will be notified when Gesture Motion start.
GCTL_SUE_STOP_MOTION	It will be notified when Gesture Motion stop.

Remarks Enqueued when the Gesture Control Device detects a power state change or a status change.

See Also "Events" on page Intro-19.

[Goto Table 1-245](#)

Device Monitor

This Chapter defines the Device Monitor device category.

Summary

Properties (UML attributes)

<i>Common</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
AutoDisable:	<i>boolean</i>	{read-write}	1.16	open
CapCompareFirmwareVersion:	<i>boolean</i>	{read-only}	1.16	open
CapPowerReporting:	<i>int32</i>	{read-only}	1.16	open
CapStatisticsReporting:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateFirmware:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateStatistics:	<i>boolean</i>	{read-only}	1.16	open
CheckHealthText:	<i>string</i>	{read-only}	1.16	open
Claimed:	<i>boolean</i>	{read-only}	1.16	open
DataCount:	<i>int32</i>	{read-only}	1.16	open
DataEventEnabled:	<i>boolean</i>	{read-write}	1.16	open
DeviceEnabled:	<i>boolean</i>	{read-write}	1.16	open & claim
FreezeEvents:	<i>boolean</i>	{read-write}	1.16	open
OutputID:	<i>int32</i>	{read-only}	1.16	Not supported
PowerNotify:	<i>int32</i>	{read-write}	1.16	open
PowerState:	<i>int32</i>	{read-only}	1.16	open
State:	<i>int32</i>	{read-only}	1.16	--
DeviceControlDescription:	<i>string</i>	{read-only}	1.16	--
DeviceControlVersion:	<i>int32</i>	{read-only}	1.16	--
DeviceServiceDescription:	<i>string</i>	{read-only}	1.16	open
DeviceServiceVersion:	<i>int32</i>	{read-only}	1.16	open
PhysicalDeviceDescription:	<i>string</i>	{read-only}	1.16	open
PhysicalDeviceName:	<i>string</i>	{read-only}	1.16	open

[Goto Table 1-246](#)

UPOS Ver1.16 RCSD Specification

Properties (Continued)

<i>Specific</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
DeviceData:	string	{read-only}	1.16	open, claim & enable
DeviceList:	string	{read-only}	1.16	open
MonitoringDeviceList:	string	{read-only}	1.16	open, claim & enable

Methods (UML operations)

Common

<i>Name</i>	<i>Version</i>
open (logicalDeviceName: string): void {raises-exception}	1.16
close (): void {raises-exception, use after open}	1.16
claim (timeout: int32): void {raises-exception, use after open}	1.16
release (): void {raises-exception, use after open, claim}	1.16
checkHealth (level: int32): void {raises-exception, use after open, enable}	1.16
clearInput (): void { }	1.16 Not supported
clearInputProperties (): void { }	1.16 Not supported
clearOutput (): void { }	Not supported
compareFirmwareVersion (firmwareFileName: string, out result: int32): void {raises-exception, use after open, enable}	1.16
directIO (command: int32, inout data: int32, inout obj: object): void {raises-exception, use after open}	1.16
resetStatistics (statisticsBuffer: string): void {raises-exception, use after open, enable}	1.16
retrieveStatistics (inout statisticsBuffer: string): void {raises-exception, use after open, enable}	1.16
updateFirmware (firmwareFileName: string): void {raises-exception, use after open, enable}	1.16
updateStatistics (statisticsBuffer: string): void {raises-exception, use after open, enable}	1.16

Specific

addMonitoringDevice (deviceID: string, monitoringMode: int32, boundary: int32, subBoundary: int32, intervalTime: int32): void {raises-exception, use after open, claim, enable}	1.16
clearMonitoringDevices (): void {raises-exception, use after open, claim, enable}	1.16
deleteMonitoringDevice (deviceID: string): void {raises-exception, use after open, claim, enable}	1.16
getDeviceValue (deviceID: string, inout *Value: int32): void {raises-exception, use after open}	1.16

[Goto Table 1-247](#)
[Goto Table 1-248](#)

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

<i>Name</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>
upos::events::DataEvent			1.16
Status:	<i>int32</i>	{read-only}	
upos::events::DirectIOEvent			1.16
EventNumber:	<i>int32</i>	{read-only}	
Data:	<i>int32</i>	{read-write}	
Obj:	<i>object</i>	{read-write}	
upos::events::ErrorEvent			1.16
ErrorCode:	<i>int32</i>	{read-only}	
ErrorCodeExtended:	<i>int32</i>	{read-only}	
ErrorLocus:	<i>int32</i>	{read-only}	
ErrorResponse:	<i>int32</i>	{read-write}	
upos::events::OutputCompleteEvent		<i>Not supported</i>	
upos::events::StatusUpdateEvent			1.16
Status:	<i>int32</i>	{read-only}	
upos::events::TransitionEvent		<i>Not supported</i>	

[Goto Table 1-249](#)

UPOS Ver1.16 RCSD Specification

General Information

The Device Monitor programmatic name is "Device Monitor".

Capabilities

The Device Monitor Device has the following capability:

- Get values measured by various devices.
- Notify the application of changes in values measured by various devices.

Device Monitor Class Diagram

The following diagram shows the relationships between the Device Monitor classes.

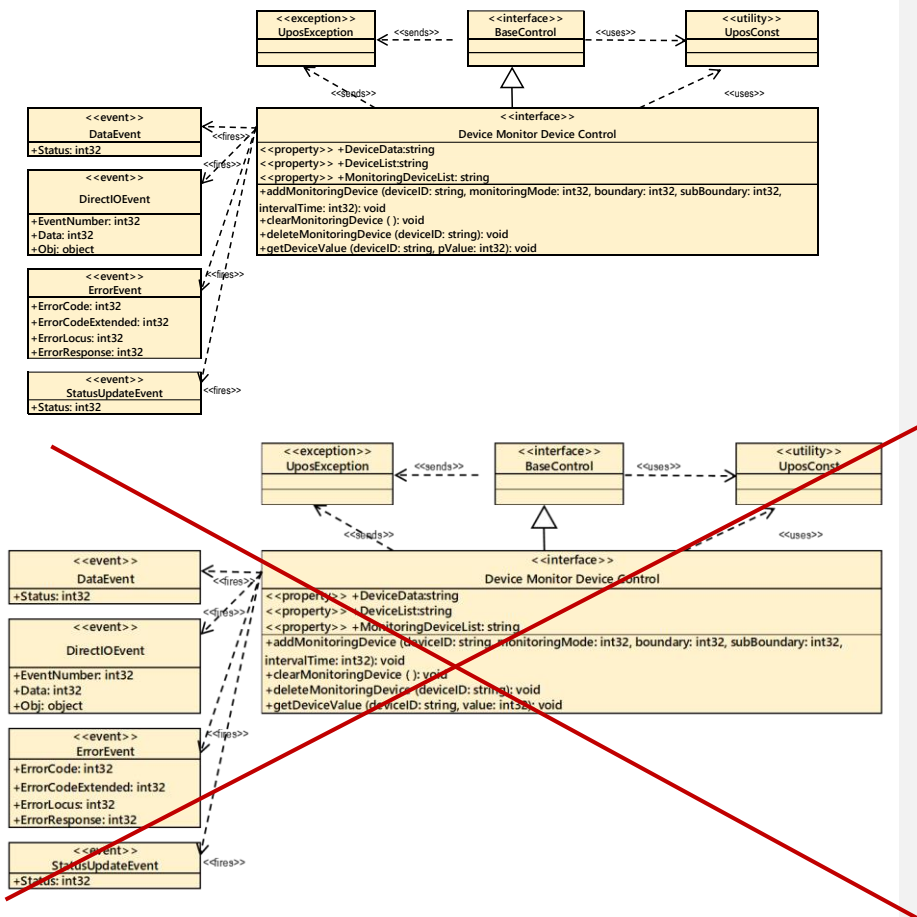


Fig. Chap. 46-1 Device Monitor Class Diagram

[Goto Table2-8](#)

Model

The Device Monitor follows the general “Device Input Model” for event-driven input:

- The Device Monitor supports monitoring of values measured by multiple devices connected to the device. A device that can be monitored and its type / value unit is listed in the **DeviceList** property.
- Device Monitor receives a change in the value measured by the device set as the monitoring target, and generates a **DataEvent** when it matches the specified condition.
- To add a device to be monitored, specify the monitoring mode with the **addMonitoringDevice** method and add it. For details on monitoring mode, see the description of **addMonitoringDevice** method.
- If the **AutoDisable** property is true, the device will automatically disable itself when a **DataEvent** is enqueued.
- An enqueued **DataEvent** can be delivered to the application when the **DataEventEnabled** property is true and other event delivery requirements are met. Just before delivering this event, data is copied into corresponding properties, and further data events are disabled by setting **DataEventEnabled** to false. This causes subsequent input data to be enqueued while the application processes the current input and associated properties. When the application has finished processing the current input and is ready for more data, it reenables events by setting **DataEventEnabled** to true.
- An **ErrorEvent** (or events) is enqueued if an error occurs while gathering or processing input, and is delivered to the application when **DataEventEnabled** is true and other event delivery requirements are met.
- The **DataCount** property can be read to obtain the total number of enqueued **DataEvents**.
- All enqueued input may be deleted by calling **ClearInput** method. See the **ClearInput** method description for more details.
- All data properties that are populated as a result of firing a **DataEvent** or **ErrorEvent** can be set back to their default values by calling the **clearInputProperties** method.
- The notified data is stored in the **DeviceData** property.
- In the **Device Monitor** device control, the measured values of the devices ~~is~~ are managed ~~with an integer value~~ most of cases with the int32 type integers, but some ~~are devices handle~~ decimals ~~values~~. In that case, the decimals are implicit, ~~you can calculate~~ and the actual value ~~can be calculated~~ by dividing the measured value by ~~the factor for each device~~ the coefficient of each device that can be ~~acquired with~~ obtained in the **DeviceList** property.
- The application will be informed about any status change with a **StatusUpdateEvent**, also, all corresponding status properties will be updated before event delivery. [Goto Table 1-250](#)

UPOS Ver1.16 RCSD Specification

Device Sharing

The Device Monitor is an exclusive-use device, as follows:

- The application must claim the device before enabling it.
- The application must claim and enable the device before the device begins reading input, or before calling methods that manipulate the device.

See the “Summary” table for precise usage prerequisites.

Properties (UML attributes)

DeviceData Property

Syntax **DeviceData:** *string* {read-only, access after open-claim-enable}

Remarks Measurement information of the device that matches the condition registered by **addMonitoringDevice** method is set.

Each measurement information consists of the following information and is shown in the following order, separated by a colon (":").

Parameter	Description
DeviceID	The target device ID.
Measured value	Measurement value of the device. The measured value is represented by an integer type. To convert it to an actual value, divide the measured value by the coefficient acquired by the DeviceList property. For example, "Device01:365" Its value is set prior to a DataEvent being delivered to the application.

Errors A UposException may be thrown when this property is accessed. For further information, see “**Errors**” on page Intro-20.

UPOS Ver1.16 RCSD Specification

DeviceList Property

Syntax **DeviceList:** *string* {read-only, access after open}

Remarks Contains the comma-delimited list of device information that are supported by the device.

Each object information consists of the following information and is shown in the following order, separated by a colon (":").

<u>Parameter</u>	<u>Description</u>
DeviceID	Indicates a unique ID in the service that identifies the device.
Type	Indicates the device type. For example, if it is a touch sensor it is expressed as "Touch Sensor" and so on. However, this value depends on the service.
Unit	Indicates the unit of value held by various devices. For example, it is expressed as "on / off" for a touch sensor, "rad / s" for a gyroscope. However, this value depends on the service.
Coefficient	Indicates the coefficient for calculating the actual measured value held by various devices. The DeviceData property and the measured value of the device that can be obtained with the GetDeviceValue method are expressed as integers, but by dividing this value by the coefficient it is the actual value. Example: Device value = 365, coefficient = 10, actual value = 36.5 For example, if one device supports one touch sensor and one gyroscope, it will be as follows. "Touch 01: Touch Sensor: ON/OFF: 1, GyroX: Gyroscope: rad/s: 100000, GyroY: Gyroscope: rad/s: 100000, GyroZ: Gyroscope: rad/s: 100000"

This property is initialized by the **open** method.

Errors A UposException may be thrown when this property is accessed. For further information, see "**Errors**" on page Intro-20.

See Also **DeviceData** Property, **addMonitoringDevice** Method, **getDeviceValue** Method.

UPOS Ver1.16 RCSD Specification

MonitoringDeviceList Property

Syntax **MonitoringDeviceList:** *string* {read-only, access after open-claim-enable}

Remarks Contains the comma-delimited list of monitoring information on registered devices that are supported by the device.

Each monitoring information consists of the following information and is shown in the following order, separated by a colon (":").

<u>Parameter</u>	<u>Description</u>
DeviceID	Registered devices ID.
Monitoring mode	Registered monitoring mode.
Boundary	Registered boundary value. This value is set to 0 when the monitoring mode does not require a boundary value.
Sub boundary	Registered sub boundary value. This value is set to 0 when the monitoring mode does not require a sub boundary value.
Interval	Registered interval. (millisecond)

For example, if you set monitoring targets as follows,

[Monitor target 1]

Device ID = Device 01, monitoring mode = DMON_MM_UPDATE,
boundary line = 0, sub boundary line = 0, interval time = 0

[Monitor target 2]

Device ID = Device 02, monitoring mode = DMON_MM_STRADDLED,
boundary line = 365, sub boundary line = 0, interval time = 500

The values shown are as follows.

"Device01:0:0:0:0, Device02:1:365:0:500"

This property is initialized by the **open** method. It is also updated by calling **addMonitoringDevice** method, **deleteMonitoringDevice** method, **clearMonitoringDevice** method.

Errors A UposException may be thrown when this property is accessed. For further information, see "**Errors**" on page Intro-20.

See Also **addMonitoringDevice** Method, **deleteMonitoringDevice** Method, **clearMonitoringDevice** Method.

[Goto Table 1-300](#)

UPOS Ver1.16 RCSD Specification
Methods (UML operations)

addMonitoringDevice Method

Syntax **addMonitoringDevice (deviceID:string, monitoringMode:int32, boundary:int32, subBoundary:int32, intervalTime:int32): void{raises-exception, use after open-claim-enable}**

Parameter	Description
deviceID	The deviceID of the monitored device. Valid values are one of the device ID lists listed in the DeviceList property.
monitoringMode	Specify the monitoring mode for monitoring.
boundary	Specify the boundary value to be monitored.
subBoundary	Specify the sub boundary value to be monitored. This value must be less than Boundary.
intervalTime	Specify the interval in milliseconds between the occurrence of the event and the start of the next monitoring.

The monitoring modes specified for MonitoringMode are as follows.

Value	Description
DMON_MMODE_UPDATE	Every time the measured value of the target device is updated, an event is notified. When set to this mode, the values of the argument boundary and subBoundary are ignored.
DMON_MMODE_STRADDLED	When the measured value of the target device crosses the value of the argument boundary, it notifies the event. In addition, when the measured value matches the value of boundary, it notifies the event even when it changes from the matched state. When set to this mode, the value of the argument subBoundary is ignored.
DMON_MMODE_HIGH	When the measured value of the target device becomes equal to or larger than the value of the argument Boundary, it notifies the event. Even if the measured value is updated and it was again equal to or greater than the value of boundary, we will notify the event will be notified in each time. When it is set to this mode, the value of the argument subBoundary is ignored.

[Goto Table 1-251](#)

UPOS Ver1.16 RCSD Specification

DMON_MMODE_LOW

Notifies the event when the measured value of the target device becomes less than or equal to the value of the argument boundary. Even when the measured value is updated and it was again less than the value of boundary, ~~we will notify~~ the event will be notified in each time. When it is set to this mode, the value of the argument subBoundary is ignored.

[Goto Table 1-252](#)

DMON_MMODE_WITHIN

It notifies the event while the measured value of the target device is within the range specified by the argument boundary and subBoundary. Even if the measured value is updated and its value is within the range again, the event is notified in each time.

DMON_MMODE_OUTSIDE

It notifies the event while the measured value of the target device is outside the range specified by the argument boundary and subBoundary. Even if the measured value is updated and its value was out of range again, ~~we will notify~~ the event will be notified in each time.

[Goto Table 1-253](#)

DMON_MMODE_POLLING

It notifies the measured value of the target device at the interval specified by intervalTime. When it is set to this mode, the values of the argument boundary and subBoundary are ignored.

Remarks Add the device specified by deviceID to the monitoring target. The monitoring mode is specified for monitoringMode, but there are monitoring modes not supported by some devices. In that case, E_ILLEGAL is raised as the UPOS exception. Devices added by this method will be added to the list of **MonitoringDeviceList** properties. If a device to be monitored is specified, it will be changed to a new condition. To exclude the added device from the monitoring target, call **deleteMonitoringDevice** method or **clearMonitoringDevice** method.

Errors A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20. Some possible values of the exception's ErrorCode property are:

Value	Description
E_ILLEGAL	An invalid value was specified, or unsupported operation with the Device

See Also **DeviceList** Property, **MonitoringDeviceList** Property, **deleteMonitoringDevice** Method, **clearMonitoringDevice** Method, **DataEvent**.

[Goto Table 1-251](#)

UPOS Ver1.16 RCSD Specification

clearMonitoringDevices Method

Syntax clearMonitoringDevices ():
void{raises-exception, use after open-claim-enable}

Remarks Exclude all devices to be monitored.

Errors A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20.

See Also addMonitoringDevice Method.

deleteMonitoringDevice Method

Syntax deleteMonitoringDevice (deviceID: string):
void{raises-exception, use after open-claim-enable}

Parameter	Description
deviceID	Specify the device ID of the device to be excluded from monitoring targets.

Remarks Exclude the device specified by deviceID from monitoring targets.

Errors A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20. Some possible values of the exception's ErrorCode property are:

Value	Description
E_ILLEGAL	An invalid value was specified, or unsupported operation with the Device.

An invalid value was specified, or unsupported operation with the Device.

See Also AddMonitoringDevice Method.

getDeviceValue method

Syntax getDeviceValue (deviceID: string, ~~inout value~~ pValue: *int32):
void{raises-exception, use after open}

Parameter	Description
deviceID	Specify the device ID of the device from which the measurement value is to be acquired. Specify one of the device ID lists listed in the DeviceList property.
pValue <i>value</i>	Measured value obtained from the device. Pointer that stores measurement values obtained from the device.

Remarks Get the measured value of the device specified by deviceID. The retrieved value is stored in pValue.

Errors A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20. Some possible values of the exception's ErrorCode property are:

Value	Description
E_ILLEGAL	An invalid value was specified, or unsupported operation with the Device.

See Also DeviceList Property. [Goto Table 1-254](#)

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

DataEvent

<<event>> upos::events::DataEvent

Status : *int32*{read-only}

Description Notifies the application when data from the Device Monitor device is available to be read.

Attributes This event contains the following attributes:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>Status</i>	<i>int32</i>	<i>Set to 0.</i>

Remarks Before this event is delivered, the individual recognition information is enqueued into the area that is indicated by the **addMonitoringDevice** method.

See Also **addMonitoringDevice** method. [Goto Table 1-255](#)

DirectIOEvent

<<event>> upos::events::DirectIOEvent

EventNumber : *int32* {read-only}

Data : *int32* {read-write}

Obj : *object* {read-write}

Description Provides Service information directly to the application. This event provides a means for a vendor-specific Device Monitor Device Service to provide events to the application that are not otherwise supported by the device control.

Attributes This event contains the following attributes:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>EventNumber</i>	<i>int32</i>	Event number whose specific values are assigned by the Service.
<i>Data</i>	<i>int32</i>	Additional numeric data. Specific values vary by the <i>EventNumber</i> and the Service. This attribute is settable.
<i>Obj</i>	<i>object</i>	Additional data whose usage varies by the <i>EventNumber</i> and the Service. This attribute is settable.

Remarks This event is to be used only for those types of vendor specific functions that are not otherwise described.

Use of this event may restrict the application program programform being used with other vendor's devices which may not have any knowledge of the Service's need for this event.

See Also "Events" on page Intro-19, **directIO** method

[Goto Table 1-256](#)

UPOS Ver1.16 RCSD Specification

ErrorEvent

<<event>> **upos::events:: ErrorEvent**
ErrorCode : *int32*{read-write}
ErrorCodeExtended : *int32*{read-write}
ErrorLocus : *int32*{read-write}
ErrorResponse : *int32*{read-write}

Description Notifies the application that a Device Monitor Device error has been detected and suitable response by the application is necessary to process the error condition.

Attributes This event contains the following attributes:

Attributes	Type	Description
<i>ErrorCode</i>	<i>int32</i>	Error code causing the error event. See a list of Error Codes on page 20.
<i>ErrorCodeExtended</i>	<i>int32</i>	Extended Error code causing the error event. If <i>ErrorCode</i> is E_EXTENDED, then see values below. Otherwise, it may contain a Service-specific value.
<i>ErrorLocus</i>	<i>int32</i>	Location of the error.
<i>ErrorResponse</i>	<i>int32</i>	Error response, whose default value may be overridden by the application (i.e., this attribute is settable). See values below.

The *ErrorLocus* attribute has one of the following values:

Value	Meaning
EL_INPUT	Error occurred while gathering or processing event-driven input. No previously buffered input data is available.
EL_INPUT_DATA	Error occurred while gathering or processing event-driven input, and some previously buffered data is available.

The application's error event handler can set the *ErrorResponse* attribute to one of the following values:

Value	Meaning
ER_RETRY	Retry sending the data. The error state is exited. May be valid for some input devices when the locus is EL_INPUT, in which case the input is re-tried, and the error state is exited.
ER_CLEAR	Valid for all loci: EL_INPUT and EL_INPUT_DATA. Clear all buffered input or output data (including all asynchronous output). The error state is exited. This is the default response when the locus is EL_INPUT.
ER_CONTINUEINPUT	Only valid when the locus is EL_INPUT_DATA. Acknowledges that a data error has occurred and directs the Device to continue input processing. The Device remains in the error state and will deliver additional DataEvents as directed by the DataEventEnabled property. When all input has been delivered and DataEventEnabled is again set to true, then another ErrorEvent is delivered with locus EL_INPUT. This is the default response when the locus is EL_INPUT_DATA.

UPOS Ver1.16 RCSD Specification

Remarks This event is enqueued when an error is detected and the Device's **State** transitions into the error state. Input error events are not delivered until **DataEventEnabled** is true, so that proper application sequencing occurs.

Unlike a **DataEvent**, the Device does not disable further **DataEvents** or input **ErrorEvents**; it leaves the **DataEventEnabled** property value at true. Note that the application may set **DataEventEnabled** to false within its event handler if subsequent input events need to be disabled for a period of time.

See Also "Device Input Model" on page Intro-22, "Error Handling" on page Intro-23,

[Goto Table 1-257](#)

StatusUpdateEvent

<<event>> **upos::events:: StatusUpdateEvent**
Status : *int32* {read-only}

Description *Notifies the application that there is an operation status change or a status of the Device Monitor device.*

Attributes This event contains the following attribute:

<u>Attributes</u>	<u>Type</u>	<u>Description</u>
<i>Status</i>	<i>int32</i>	Indicates a change in the Device Monitor status of the unit.

*Note that Release 1.3 added Power State Reporting with additional Power reporting **StatusUpdateEvent** values.*

The Update Firmware capability added additional *Status* values for communicating the status/progress of an asynchronous update firmware process. See "**StatusUpdateEvent**" description on page 1-34.

<u>Value</u>	<u>Meaning</u>
--------------	----------------

DMON_SUE_START_MONITORING
It will be notified when Device Monitoring start.

DMON_SUE_STOP_MONITORING
It will be notified when Device Monitoring stop.

Remarks Enqueued when the Device Monitor Device detects a power state change or a status change.

See Also "Events" on page Intro-19.

[Goto Table 1-258](#)

Graphic Display

This Chapter defines the Graphic Display device category.

Summary

Properties (UML attributes)

<i>Common</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
AutoDisable:	<i>boolean</i>	{read-write}	1.16	open Not supported
CapCompareFirmwareVersion:	<i>boolean</i>	{read-only}	1.16	open
CapPowerReporting:	<i>int32</i>	{read-only}	1.16	open
CapStatisticsReporting:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateFirmware:	<i>boolean</i>	{read-only}	1.16	open
CapUpdateStatistics:	<i>boolean</i>	{read-only}	1.16	open
CheckHealthText:	<i>string</i>	{read-only}	1.16	open
Claimed:	<i>boolean</i>	{read-only}	1.16	open
DataCount:	<i>int32</i>	{read-only}	1.16	open Not supported
DataEventEnabled:	<i>boolean</i>	{read-write}	1.16	open Not supported
DeviceEnabled:	<i>boolean</i>	{read-write}	1.16	open, & claim
FreezeEvents:	<i>boolean</i>	{read-write}	1.16	open
OutputID:	<i>int32</i>	{read-only}	1.16	open
PowerNotify:	<i>int32</i>	{read-write}	1.16	open
PowerState:	<i>int32</i>	{read-only}	1.16	open
State:	<i>int32</i>	{read-only}	1.16	--
DeviceControlDescription:	<i>string</i>	{read-only}	1.16	--
DeviceControlVersion:	<i>int32</i>	{read-only}	1.16	--
DeviceServiceDescription:	<i>string</i>	{read-only}	1.16	open
DeviceServiceVersion:	<i>int32</i>	{read-only}	1.16	open
PhysicalDeviceDescription:	<i>string</i>	{read-only}	1.16	open
PhysicalDeviceName:	<i>string</i>	{read-only}	1.16	open

[Goto Table 1-259](#)

UPOS Ver1.16 RCSD Specification

Properties (Continued)

<i>Specific</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>	<i>May Use After</i>
CapAssociatedHardTotalsDevice:	<i>string</i>	{read-only}	1.16	open
CapBrightness:	<i>boolean</i>	{read-only}	1.16	open
CapImageType:	<i>boolean</i>	{read-only}	1.16	open
CapStorage:	<i>int32</i>	{read-only}	1.16	open
CapURLBack:	<i>boolean</i>	{read-only}	1.16	open
CapURLForward:	<i>boolean</i>	{read-only}	1.16	open
CapVideoType:	<i>boolean</i>	{read-only}	1.16	open
CapVolume:	<i>boolean</i>	{read-only}	1.16	open
Brightness:	<i>int32</i>	{read-write}	1.16	open, claim & enable
DisplayMode:	<i>int32</i>	{read-write}	1.16	open, claim & enable
ImageType:	<i>string</i>	{read-write}	1.16	open, claim & enable
CapImageTypeList:	<i>string</i>	{read-only}	1.16	open
LoadStatus:	<i>int32</i>	{read-only}	1.16	open
Storage:	<i>int32</i>	{read-write}	1.16	open, claim & enable
URL:	<i>string</i>	{read-only}	1.16	open
VideoType:	<i>string</i>	{read-write}	1.16	open, claim & enable
CapVideoTypeList:	<i>string</i>	{read-only}	1.16	open
Volume:	<i>int32</i>	{read-write}	1.16	open, claim & enable

Methods (UML operations)

Common

<i>Name</i>	<i>Version</i>
open (logicalDeviceName: <i>string</i>): void {raises-exception}	1.16
close (): void {raises-exception, use after open}	1.16
claim (timeout: <i>int32</i>): void {raises-exception, use after open}	1.16
release (): void {raises-exception, use after open, claim}	1.16
checkHealth (level: <i>int32</i>): void {raises-exception, use after open, enable}	1.16
clearInput (): void { }	Not Supported 1.16
clearInputProperties (): void { }	Not Supported 1.16

[Goto Table 1-260](#)

[Goto Table 1-261](#)

[Goto Table 1-262](#)

[Goto Table 1-298](#)

UPOS Ver1.16 RCSD Specification

Methods (UML operations)(Continued)

clearOutput (): void { }	1.16
compareFirmwareVersion (firmwareFileName: string, out result: int32): void {raises-exception, use after open, enable}	1.16
directIO (command: int32, inout data: int32, inout obj: object): void {raises-exception, use after open}	1.16
resetStatistics (statisticsBuffer: string): void {raises-exception, use after open, enable}	1.16
retrieveStatistics (inout statisticsBuffer: string): void {raises-exception, use after open, enable}	1.16
updateFirmware (firmwareFileName: string): void {raises-exception, use after open, enable}	1.16
updateStatistics (statisticsBuffer: string): void {raises-exception, use after open, enable}	1.16

Specific

<i>Name</i>	<i>Version</i>
cancelURLLoading (): void {raises-exception, use after open, claim, enable}	1.16
goURLBack (): void {raises-exception, use after open, claim, enable}	1.16
goURLForward (): void {raises-exception, use after open, claim, enable}	1.16
loadImage (fileName: string): void {raises-exception, use after open, claim, enable}	1.16
loadURL (uURL: string): void {raises-exception, use after open, claim, enable}	1.16
playVideo (fileName: string, loop: boolean): void { raises-exception, use after open, claim, enable}	1.16
stopVideo (): void {raises-exception, use after open, claim, enable}	1.16
updateURLPage (): void {raises-exception, use after open, claim, enable}	1.16

[Goto Table 1-263](#)

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

<i>Name</i>	<i>Type</i>	<i>Mutability</i>	<i>Version</i>
upos::events::DataEvent			
Status:		{read-only} <i>Not supported</i>	
upos::events::DirectIOEvent			
			1.16
EventNumber:	<i>int32</i>	{read-only}	
Data:	<i>int32</i>	{read-write}	
Obj:	<i>object</i>	{read-write}	
upos::events::ErrorEvent			
			1.16
ErrorCode:	<i>int32</i>	{read-only}	
ErrorCodeExtended:	<i>int32</i>	{read-only}	
ErrorLocus:	<i>int32</i>	{read-only}	
ErrorResponse	<i>int32</i>	{read-write}	
upos::events::OutputCompleteEvent			
			1.16
OutputID:	<i>int32</i>	{read-only}	
upos::events::StatusUpdateEvent			
			1.16
Status:	<i>int32</i>	{read-only}	
upos::events::TransitionEvent		<i>Not supported</i>	

[Goto Table 1-264](#)
[GotoTable 1-299](#)

UPOS Ver1.16 RCSD Specification

General Information

The Graphic Display programmatic name is “Graphic Display”.

Capabilities

The Graphic Display has the following capability:

- Displays the specified image files.
- Play the specified video.
- Display the specified web page.
- Notify the application of changes in the load status of the web page.

UPOS Ver1.16 RCSD Specification

Graphics Display Class Diagram

The following diagram shows the relationships between the Graphic Display classes.

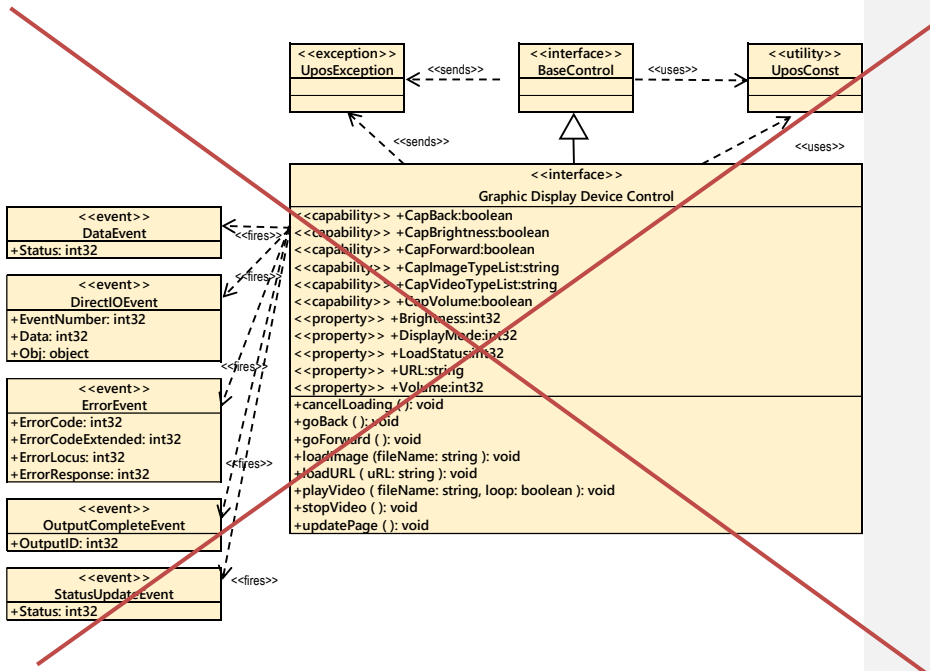
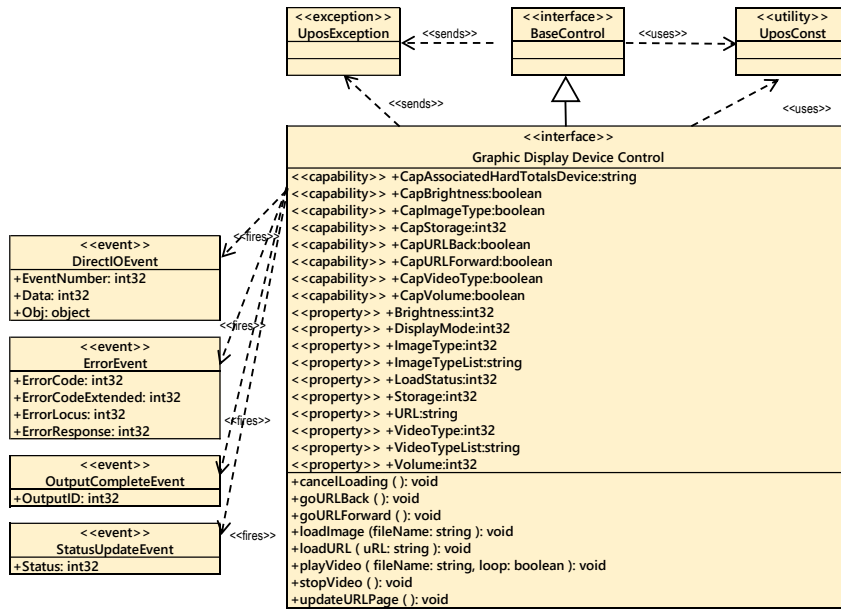


Fig. Chap. 47-1 Graphic Display Class Diagram

[Goto Table 2-9](#)

UPOS Ver1.16 RCSD Specification

Model

The following display modes exist in the graphics control, and the model differs depending on the display mode:

- Image display mode
- ~~Movie~~Video display mode.
- Web display mode.

The application can change the display mode by changing the value of the **DisplayMode** property.

[Goto Table 1-265](#)

Image Display Mode

The image display mode of the graphics control is as follows.

~~The application calls the **loadImage** method to display the image. The **CapImageTypeList** property lists image files that the device can display. Applications need to support “hard total” services as image files displaying with **loadImage** method must be placed in the area managed by the “hard total” service~~

The application calls the **loadImage** method to display the image. The **CapImageTypeList** property lists image files that the device can display. Applications need to support “hard total” services as image files displaying with **loadImage** method must be placed in the area managed by the “hard total” service.

Prior to start this mode, need to set the appropriate image type file value in the **ImageType** property from the listed values in the **ImageTypeList** property, if **CapImageType** property is true. Then the application can call the **loadImage** method to display the image. Raises **StatusUpdateEvent** at the status change timing of image load start with status GDSP_SUE_START_IMAGE_LOAD, and image load end with status GDSP_SUE_END_IMAGE_LOAD. The **ImageTypeList** property lists image files that the device can display.

Applications may need to support “**Hard Totals**” services as image files displaying with **loadImage** method might be placed in the area managed by the associated “**Hard Totals**” service device. If the **CapStorage** is either GDSP_CST_ALL or GDSP_CST_HARDTOTALS_ONLY, it is possible to store it in the Associated Hard Totals device and storage device’s open name is held in the **CapAssociatedHardTotalsDevice** property.

If device supports both Hard Totals device and the host file system, the application should set the **Storage** property accordingly to tell where to write the image data file.

[Goto Table 1-266](#)

Movie-Video Display Mode

The video display mode of Graphic Display follows the general device behavior model for asynchronous output devices.

The graphics control of video display modes are as follows.

~~The application calls a **playVideo** method to start playing video.~~

Prior to start this mode, need to set the appropriate video type file value in the **VideoType** property from the listed values in the **VideoTypeList** property, if **CapVideoType** property is true.

Then the application can call the **playVideo** method to display the video. Also, the video being displayed is stopped by calling the **stopVideo** method.

Raises **StatusUpdateEvent** at the status change timing of start play video with status GDSP_SUE_START_PLAY_VIDEO, and stop play video with status GDSP_SUE_STOP_PLAY_VIDEO.

The Device validates the method parameters an error condition immediately if

UPOS Ver1.16 RCSD Specification

necessary. If the validation is successful, the Device does the following:

- ~~1~~ Buffers the request in program memory, for delivery to the Physical Device as soon as the Physical Device can receive and process it.
- ~~2~~ Sets the **OutputID** property to a unique integer identifier for this request.
- ~~3~~ Returns as soon as possible.

When the Device successfully completes a request, an **OutputCompleteEvent** is enqueued for delivery to the application.

A property of this event contains the output ID of the completed request.

The application should compare the returned **OutputCompleteEvent** property **OutputID** value with the **OutputID** value set by the asynchronous process method call used to send the data in order to track what data has been successfully sent to the device.

If an error occurs while processing a request, an **ErrorEvent** is enqueued which will be delivered to the application after the events already enqueued, including **OutputCompleteEvents**. No further asynchronous output will occur until the event has been delivered to the application. If the response is ER_CLEAR, then outstanding asynchronous output is cleared.

If the response is ER_RETRY, then output is retried; note that if several outputs were simultaneously in progress at the time that the error was detected, then the Service may need to retry all of these outputs.

Asynchronous output is always performed on a first-in first-out basis. If the device supports concurrent playback, the request will be executed simultaneously. To check if the device supports simultaneous playback, check the **CapMultiPlay** property.

If the request is terminated before completion, due to reasons such as the application calling the **clearOutput** method, then no **OutputCompleteEvent** is delivered. It can also delete the output individually by calling the **stopVideo** method. Also, in this case **OutputCompleteEvent** will not be notified.

~~The **CapVideoTypeList** property lists video files that the device can play. The video files that the device can display are listed in the **VideoTypeList** property. Applications need to support "hard total" services as video files played with the **playVideo** method must be placed in the area managed by the "hard total" service.~~

Since video files to be displayed using the **playVideo** method must be placed in an area managed by the associated "**Hard Totals**" service device. If the **CapStorage** is either GDSP_CST_ALL or GDSP_CST_HARDTOTALS_ONLY, it is possible to store it in the Associated Hard Totals device and storage device's open name is held in the **CapAssociatedHardTotalsDevice** property.

If device supports either or both Hard Totals device and the host file system, the application should set the **Storage** property accordingly to tell where to write the image data file.

The video display mode of graphics control follows an asynchronous output model. Raises **StatusUpdateEvent** if Graphic Display device power status or a device status changes are occurred during the video displaying.

[Goto Table 1-267](#)

UPOS Ver1.16 RCSD Specification

Web Display Mode

~~The web display mode of the Graphics Display follows the general “Device Input Model” for event driven input.~~

~~When input is received from the Graphics Display, a **DataEvent** is enqueued.~~

~~If the **AutoDisable** property is true, then the device automatically disables itself when a **DataEvent** is enqueued.~~

~~An enqueued **DataEvent** can be delivered to the application when the **DataEventEnabled** property is true and other event delivery requirements are met. Just before delivering this event, data is copied into corresponding properties, and further data events are disabled by setting **DataEventEnabled** to false.~~

~~This causes subsequent input data to be enqueued while the application processes the current input and associated properties. When the application has finished processing the current input and is ready for more data, it reenables events by setting **DataEventEnabled** to true.~~

~~An **ErrorEvent** (or events) is enqueued if an error occurs while gathering or processing input, and is delivered to the application when **DataEventEnabled** is true and other event delivery requirements are met.~~

~~The **DataCount** property may be read to obtain the total number of enqueued **DataEvents**.~~

~~All enqueued input may be deleted by calling **clearInput**. See the **clearInput** method description for more details.~~

~~All data properties that are populated as a result of firing a **DataEvent** or **ErrorEvent** can be set back to their default values by calling the **clearInputProperties** method.~~

~~The load state of the web page is stored in the **LoadStatus** property, and the URL is stored in the **URL** property.~~

The web display mode of graphics control is as follows.

The application calls the **loadURL** method to display the web page.

Raises **StatusUpdateEvent** at the timing of Web page load start with status **GDSP_SUE_START_LOAD_WEBPAGE**, load finish with status **GDSP_SUE_FINISH_LOAD_WEBPAGE**, and load cancel with status **GDSP_SUE_CANCEL_LOAD_WEBPAGE**. And application can detect the web page loading status.

The latest loading status of the web page is stored in the **LoadStatus** property when **loadURL** method is called, and its URL information is stored in the **URL** property.

In case when **cancelLoading** method is called during the loading process, current accessed URL information will be stored in the **URL** property.

The graphics control web display mode follows an asynchronous output model.

[Goto Table 1-268](#)

Device Sharing

The Graphic Display Device is an exclusive-use device, as follows:

- The application must claim the device before enabling it.
- The application must claim and enable the device before accessing some properties or calling methods that update the device.

See the “Summary” table for precise usage prerequisites.

UPOS Ver1.16 RCSD Specification

Properties (UML attributes)

Brightness Property

Syntax	Brightness: <i>int32</i> {read-write, access after open-claim-enable}				
Remarks	Holds the brightness of screen. Legal values range from zero through 100. This property is initialized by the open method.				
Errors	A UposException may be thrown when this method is invoked. For further information, see “ Errors ” on page Intro-20. Some possible values of the exception’s <i>ErrorCode</i> property are: <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified.</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	E_ILLEGAL	An invalid value was specified.
<u>Value</u>	<u>Meaning</u>				
E_ILLEGAL	An invalid value was specified.				

See Also CapBrightness Property.

CapAssociatedHardTotalsDevice Property

Syntax	CapAssociatedHardTotalsDevice : <i>string</i> {read-write, access after open}
Remarks	Holds the open name of the associated Hard Totals device if the device is able to write to such devices which is the case if CapStorage is either GDSP_CST_ALL or GDSP_CST_HARDTOTALS_ONLY. If CapStorage is GDSP_CST_HOST_ONLY this property value must be the empty string.
Errors	UposException may be thrown when this property is accessed. For further information, see “ Errors ” on page Intro-20.
See Also	CapStorage Property Goto Table 1-269

CapBrightness Property

Syntax	CapBrightness: <i>boolean</i> {read-only, access after open}
Remarks	If true, the application can change the screen brightness. If false, the application cannot change the screen brightness. This property is initialized by the open method.
Errors	A UposException may be thrown when this method is invoked. For further information, see “ Errors ” on page Intro-20.
See Also	Brightness Property.

CapImageType Property

Syntax	CapImageType: <i>boolean</i> {read-only, access after open}
Remarks	If true, indicate the image type file to be used in this target device as the value of the ImageType property. Otherwise it is false. This property is initialized by the open method.
Errors	A UposException may be thrown when this method is invoked. For further information, see “ Errors ” on page Intro-20.
See Also	ImageType Property, ImageTypeList Property

[Goto Table 1-270](#)

UPOS Ver1.16 RCSD Specification

CapStorage Property

Syntax **CapStorage:** *int32* {read-only, access after open}

Remarks This is an enumeration and announces where the device is able to write the image data file to.
It holds one of the following values.

<u>Value</u>	<u>Meaning</u>
GDSP_CST_HARDTOTALS_ONLY	Only an associate Hard Totals device is supported.
GDSP_CST_HOST_ONLY	Only the host's file system is supported.
GDSP_CST_ALL	Both, the associated Hard Totals device and the host's file system is supported.

This property is initialized by the **open** method.

If a Hard Totals device is supported the Storage the property value should be GDSP_CST_HARDTOTALS_ONLY or GDSP_CST_ALL, and the property **CapAssociatedHardTotalsDevice** holds the open name of the associated **Hard Totals** device.

Errors UposException may be thrown when this property is accessed.
For further information, see "Errors" on page Intro-20.

See Also **Storage Property, CapAssociatedHardTotalsDevice Property**

[Goto Table 1-271](#)

CapURLBack Property

Syntax **CapURLBack:** *boolean* {read-only, access after open}

Remarks If true, the previous page exists in the browsing history. Application can return to the previous page with **goURLBack** method.

If false, there is no previous page in the browsing history.

This property is initialized to false by the open method. Also, as the web page loading state changes, it is set by the **device** control.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.

See Also **goURLBack Method.**

[Goto Table 1-272](#)

CapURLForward Property

Syntax **CapURLForward:** *boolean* {read-only, access after open}

Remarks If true, the next page exists in the browsing history. Application can go to the next page with the **goURLForward** method.

If false, there is no next page in the browsing history.

This property is initialized to false by the open method. Also, as the web page loading state changes, it is set by the **device** control.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.

See Also **goURLForward Method.**

[Goto Table 1-273](#)

UPOS Ver1.16 RCSD Specification

CapVideoType Property

- Syntax** **CapVideoType:** *boolean* {read-only, access after open}
- Remarks** If true, indicate the vide type value that can be used in this targeted graphics display device as the value of VideoType Property. Otherwise, it is false. This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.
- See Also** **VideoType** Property, **VideoTypeList** Property .

[Goto Table 1-274](#)

CapVolume Property

- Syntax** **CapVolume:** *boolean* {read-only, access after open}
- Remarks** If true, the application can change the volume of video.
If false, the application cannot change the volume of video.
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.
- See Also** **Volume** Property.

UPOS Ver1.16 RCSD Specification

DisplayMode Property

Syntax **DisplayMode:** *int32* {read-write, access after open-claim-enable}

Remarks Holds the image and/or video displaying mode.

<u>Value</u>	<u>Meaning</u>
--------------	----------------

GD ISP _DMODE_HIDDEN	
---------------------------------	--

~~Hide the screen.~~ It is a mode to hide images and/or video

GD ISP _DMODE_IMAGE_FIT	
------------------------------------	--

It is a mode to display images. The displayed image is enlarged / reduced to the size that maintains the aspect and ~~just enter~~ fits on the screen.

GD ISP _DMODE_IMAGE_FILL	
-------------------------------------	--

It is a mode to display images.
The displayed image is scaled to the size that maintains the aspect and covers the entire screen.

GD ISP _DMODE_IMAGE_CENTER	
---------------------------------------	--

It is a mode to display images.
The displayed image is displayed in the center of the screen without changing the size.

GD ISP _DMODE_VIDEO_NORMAL	
---------------------------------------	--

It is a mode to display video. The displayed ~~movie~~ video will be displayed in the center of the screen without resizing.

GD ISP _DMODE_VIDEO_FULL	
-------------------------------------	--

It is a mode to display video.
The displayed video will be displayed in full screen.

GD ISP _DMODE_WEB	
------------------------------	--

Display the web screen.

If application hide other modes and screens while displaying images, ~~movies~~ videos, or web, all displayed contents will be cleared. The ~~movie~~ video will be stopped while the ~~movie~~ video is playing.

This property is initialized by the **open** method.

Errors A UpoxException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.

Some possible values of the exception's *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
--------------	----------------

E_ILLEGAL	An invalid value was specified.
-----------	---------------------------------

See Also **CapCaptureColorSpaceList** Property, **VideoCaptureMode** Property
~~readFrame Method~~ (They are Video Capture Device Properties)

[Goto Table 1-275](#)

UPOS Ver1.16 RCSD Specification

ImageType Property

- Syntax** **ImageType:** *string* {read-write, access after open-claim-enable}
- Remarks** Contains the image file type that are support by the device, if **CapImageType** property is true. For example, if the device supports BMP, then this property should be set to "BMP". This property value should be set prior to execute the **loadImage** method. All of the capable image file types are listed in the **ImageTypeList** property. *Notation contents may be different depending on the device. This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.
- See Also** **CapImageType** Property, **ImageTypeList** Property, **loadImage** Method.

[Goto Table 1-276](#)

CapImageTypeList Property

- Syntax** **CapImageTypeList:** *string* {read-only, access after open}
- Remarks** Contains the comma-delimited list of image file type that are support by the device. For example, if the device only supports BMP and JPEG, then this property should be set to "BMP,JPEG". One of value in the property should be set in the **ImageType** property, if **CapImageType** property is true, prior to execute the **loadImage** method.
- *Notation contents may be different depending on the device.
- This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.
- See Also** **CapImageType** Property, **ImageType** Property, **loadImage** Method.

[Goto Table 1-277](#)

LoadStatus Property

- Syntax** **LoadStatus:** *int32* {read-only, access after open-claim-enable}
- Remarks** Holds loading state of web page.
- The parameters to be set are as follows.
- | <u>Value</u> | <u>Meaning</u> |
|-----------------------------------|---|
| GD ISP _LSTATUS_START | Start loading the web page. |
| GD ISP _LSTATUS_FINISH | It has ve finished loading the web page. |
| GD ISP _LSTATUS_CANCEL | It has ve canceled loading the web page |
- Its value is set prior to a **StatusUpdateDataEvent** being delivered to the application.
- Errors** A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.

[Goto Table 1-278](#)

UPOS Ver1.16 RCSD Specification

Storage Property

Syntax	Storage: <i>int32</i> {read-write, access after open-claim-enable}								
Remarks	This is an enumeration and defines where the device writes the recorded image data file to. Should be set before an appropriate method call. It holds one of the following values. <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>GDSP_ST_HARDTOTALS</td><td>The image data file is written to the associated Hard Totals device. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device.</td></tr><tr><td>GDSP_ST_HOST</td><td>The image data file is written to the host's file system.</td></tr><tr><td>GDSP_ST_HOST_HARDTOTALS</td><td>The encoded data file is written to the associated Hard Totals device and host's file system. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device.</td></tr></tbody></table> <p>This property is initialized by the open method according to the value hold by CapStorage. If CapStorage has the value GDSP_CST_ALL, it is initialized to GDSP_ST_HOST_HARDTOTALS.</p>	<u>Value</u>	<u>Meaning</u>	GDSP_ST_HARDTOTALS	The image data file is written to the associated Hard Totals device. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device.	GDSP_ST_HOST	The image data file is written to the host's file system.	GDSP_ST_HOST_HARDTOTALS	The encoded data file is written to the associated Hard Totals device and host's file system. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device.
<u>Value</u>	<u>Meaning</u>								
GDSP_ST_HARDTOTALS	The image data file is written to the associated Hard Totals device. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device.								
GDSP_ST_HOST	The image data file is written to the host's file system.								
GDSP_ST_HOST_HARDTOTALS	The encoded data file is written to the associated Hard Totals device and host's file system. The property CapAssociatedHardTotalsDevice holds the open name of the associated Hard Totals device.								
Errors	UposException may be thrown when this property is accessed. For further information, see "Errors" on page Intro-20. <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>E_ILLEGAL</td><td>An invalid value was specified, or recording is ongoing.</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	E_ILLEGAL	An invalid value was specified, or recording is ongoing.				
<u>Value</u>	<u>Meaning</u>								
E_ILLEGAL	An invalid value was specified, or recording is ongoing.								
See Also	CapStorage Property, CapAssociatedHardTotalsDevice Property								

[Goto Table 1-279](#)

URL Property

Syntax	URL: <i>string</i> {read-only, access after open-claim-enable}
Remarks	When the LoadStatus property is GD I SP_LSTATUS_START, the URL of the Web page that starts loading is set. When the LoadStatus property is GD I SP_LSTATUS_FINISH, the URL of the loaded Web page is set. When the LoadStatus property is GD I SP_STATUS_CANCEL, the URL of the canceled Web page is set. Its value is set prior to a StatusUpdateDataEvent being delivered to the application.
Errors	A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20.
See Also	loadStatus Property. Goto Table 1-280

UPOS Ver1.16 RCSD Specification

VideoType Property

- Syntax** **VideoType:** *string* {read-write, access after open-claim-enable}
- Remarks** Contains the video file type that are support by the device, if **CapVideoType** property is true. For example, if the device supports AVI MJPG, then this property should be set to “AVI MJPG”. This property value should be set prior to execute the **playVideo** method. All of the capable video file types are listed in the **VideoTypeList** property.
*Notation contents may be different depending on the device.
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.
- See Also** **CapVideoType** Property, **VideoTypeList** Property, **playVideo** Method.

[Goto Table 1-281](#)

CapVideoTypeList Property

- Syntax** **CapVideoTypeList:** *string* {read-only, access after open}
- Remarks** Contains the comma-delimited list of video file type that are support by the device. if the device only supports AVI_IYUV and AVI_MJPG, then this property should be set to “AVI_IYUV,AVI_MJPG”. [One of value in the property should be set in the VideoType property, if CapImageType property is true, prior to execute the playVideo method.](#)
*Notation contents may be different depending on the device.
This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.
- See Also** **CapVideoType** Property, **VideoType** Property, **playVideo** Method.

[Goto Table 1-282](#)

Volume Property

- Syntax** **Volume:** *int32* {read-write, access after open-claim-enable}
- Remarks** Holds the volume at playing video. Legal values range from zero through 100. This property is initialized by the **open** method.
- Errors** A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.
Some possible values of the exception’s **ErrorCode** property are:
- | <u>Value</u> | <u>Meaning</u> |
|--------------|---------------------------------|
| E_ILLEGAL | An invalid value was specified. |
- See Also** **CapVolume** Property, **playVideo** Method.

UPOS Ver1.16 RCSD Specification

Methods (UML operations)

cancelURLLoading Method

Syntax **cancelURLLoading ():**
 void {raises-exception, use after open-claim-enable}

Remarks Cancel loading web page.
 This method is executed asynchronously. The load status is reported by **StatusUpdateDataEvent** and **OutputCompleteEvent** or **ErrorEvent**.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.
 Some possible values of the exception’s *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	It is not loading.

[Goto Table 1-283](#)

goURLBack Method

Syntax **goURLBack ():**
 void {raises-exception, use after open-claim-enable}

Remarks It returns to the previous page of browsing history.
 This method is executed asynchronously. The load status is reported by **StatusUpdateDataEvent** and **OutputCompleteEvent** or **ErrorEvent**.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.
 Some possible values of the exception’s *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	There is no previous page in the browsing history.

See Also **CapURLBack** Property.

[Goto Table 1-284](#)

goURLForward Method

Syntax **goURLForward ():**
 void {raises-exception, use after open-claim-enable}

Remarks Go to the next page of browsing history.
 This method is executed asynchronously. The load status is reported by **StatusUpdateDataEvent** and **OutputCompleteEvent** or **ErrorEvent**.

Errors A UposException may be thrown when this method is invoked. For further information, see “**Errors**” on page Intro-20.
 Some possible values of the exception’s *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	There is no next page in the browsing history.

See Also **CapURLForward** Property.

[Goto Table 1-285](#)

UPOS Ver1.16 RCSD Specification

loadImage Method

Syntax **loadImage (fileName: string):**
 void {raises-exception, use after open-claim-enable}

<u>Parameter</u>	<u>Description</u>
<i>fileName</i>	Specify the file name of the image to be loaded.

Remarks Load the specified image.

This method fails if the value of the **DisplayMode** Property is not set to GD*SP_DMODE_IMAGE_FIT, GD*SP_DMODE_IMAGE_FILL, or GD*SP_DMODE_IMAGE_CENTER.

Image files ~~must be~~ are located in the area ~~managed by "Hard Totals" service~~ as the stored values of the **Storage** property.

This method is executed asynchronously. Image file loading status is reported by **StatusUpdateEvent** and **OutputCompleteEvent** or **ErrorEvent**.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.

Some possible values of the exception's **ErrorCode** property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified. Or an unsupported image file was specified.
E_NOEXIST	File does not exist.

See Also **DisplayMode** Property. [Goto Table 1-286](#)

loadURL Method

Syntax **loadURL (uRL: string):**
 void {raises-exception, use after open-claim-enable}

<u>Parameter</u>	<u>Description</u>
<i>uRL</i>	Specify the uRL of the web page to load.

Remarks Load the web page with the specified ~~u~~URL.

This method is executed asynchronously. The load status is reported by **StatusUpdateDataEvent** and **OutputCompleteEvent** or **ErrorEvent**.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.

Some possible values of the exception's **ErrorCode** property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified.

[Goto Table 1-287](#)

UPOS Ver1.16 RCSD Specification

playVideo Method

Syntax **playVideo** (*fileName: string, loop: boolean*):
 void {raises-exception, use after open-claim-enable}

<u>Parameter</u>	<u>Description</u>
<i>fileName</i>	Specify the file name of the video to be played.
<i>loop</i>	If true, loop playback is performed, and if false, loop playback is not performed.

Remarks Play the specified video that are loaded in the storage area by the **loadImage** method. All of loaded file images are listed in the **ImageTypeList** property.

If the value of the **DisplayMode** property is not set to GDISP_DMODE_VIDEO_NORMAL, GDISP_DMODE_VIDEO_FULL, this method will fail.

This method is executed asynchronously. To stop video displaying in the middle, call the **stopVideo** method.

Video files are ~~must be~~ located in the area ~~managed by "Hard Totals" service~~ as the stored values of the **Storage** property.

The video file playing status will be informed by the **StatusUpdateEvent**.

This method is executed asynchronously. Image file loading status and video file playing status are reported by **StatusUpdateEvent** and **OutputCompleteEvent** or **ErrorEvent**.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.

Some possible values of the exception's **ErrorCode** property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	An invalid value was specified. Or an unsupported video file was specified.
E_NOEXIST	File does not exist.

See Also **DisplayMode** Property.

[Goto Table 1-288](#)

stopVideo Method

Syntax **stopVideo** (**void**):
 void {raises-exception, use after open-claim-enable}

Remarks Stop the video being displayed.

This method is executed asynchronously. Image file loading status is reported by **StatusUpdateEvent** and **OutputCompleteEvent** or **ErrorEvent**.

Errors A UposException may be thrown when this method is invoked. For further information, see "**Errors**" on page Intro-20.

Some possible values of the exception's **ErrorCode** property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	The Video is not playing.

See Also **startVideo** Method.

[Goto Table 1-289](#)

UPOS Ver1.16 RCSD Specification

updateURLPage Method

Syntax **updateURLPage ():**
 void {raises-exception, use after open-claim-enable}

Remarks Reload the current web page.

This method is executed asynchronously. The load status is reported by **StatusUpdateDataEvent** and **OutputCompleteEvent** or **ErrorEvent**.

Errors A UposException may be thrown when this method is invoked. For further information, see "Errors" on page Intro-20.

Some possible values of the exception's *ErrorCode* property are:

<u>Value</u>	<u>Meaning</u>
E_ILLEGAL	Web page loading.

[Goto Table 1-290](#)

UPOS Ver1.16 RCSD Specification

Events (UML interfaces)

DirectIOEvent

<<event>> **upos::events::DirectIOEvent**

EventNumber : *int32* {read-only}
Data : *int32* {read-write}
Obj : *object* {read-write}

Description Provides Service information directly to the application. This event provides a means for a vendor-specific Sound Player Service to provide events to the application that are not otherwise supported by the device control.

Attributes This event contains the following attributes:

<u>Attribute</u>	<u>Type</u>	<u>Description</u>
<i>EventNumber</i>	<i>int32</i>	Event number whose specific values are assigned by the Service.
<i>Data</i>	<i>int32</i>	Additional numeric data. Specific values vary by the <i>EventNumber</i> and the Service. This attribute is settable.
<i>Obj</i>	<i>object</i>	Additional data whose usage varies by the <i>EventNumber</i> and the Service. This attribute is settable.

Remarks This event is to be used only for those types of vendor specific functions that are not otherwise described.
 Use of this event may restrict the application program programform being used with other vendor's devices which may not have any knowledge of the Service's need for this event.

See Also "Events" on page Intro-19, **directIO** method

[Goto Table 1-291](#)

ErrorEvent

<<event>> **upos::events:: ErrorEvent**

ErrorCode : *int32*{read-write}
ErrorCodeExtended : *int32*{read-write}
ErrorLocus : *int32*{read-write}
ErrorResponse : *int32*{read-write}

Description Notifies the application that a Graphic Display Device error has been detected and suitable response by the application is necessary to process the error condition.

Attributes This event contains the following attributes:

<u>Attributes</u>	<u>Type</u>	<u>Description</u>
<i>ErrorCode</i>	<i>int32</i>	Error code causing the error event. See a list of Error Codes on page 20.
<i>ErrorCodeExtended</i>	<i>int32</i>	Extended Error code causing the error event. If <i>ErrorCode</i> is E_EXTENDED, then see values below. Otherwise, it may contain a Service-specific value.
<i>ErrorLocus</i>	<i>int32</i>	Location of the error. If EL_OUTPUT is specified. An error occurred during asynchronous action.
<i>ErrorResponse</i>	<i>int32</i>	Error response, whose default value may be overridden by the application (i.e., this attribute is settable). See values below.

UPOS Ver1.16 RCSD Specification

If `ErrorCode` is `E_EXTENDED`, then `ErrorCodeExtended` has one of the following values:

Value	Meaning
<code>EGDSP_NOROOM</code>	There is not enough room to store the targeted device for the image data file.

The `ErrorLocus` attribute has one of the following values:

Value	Meaning
<code>EL_OUTPUT</code>	Error occurred while processing asynchronous output.

The application's error event handler can set the `ErrorResponse` attribute to one of the following values:

Value	Meaning
<code>ER_RETRY</code>	Retry sending the data. The error state is exited. Typically, valid for asynchronous output devices when the locus is <code>EL_OUTPUT</code> , in which case the asynchronous output is retried and the error state is exited. This is the default response when the locus is <code>EL_OUTPUT</code> .
<code>ER_CLEAR</code>	Valid for loci: <code>EL_OUTPUT</code> . Clear all buffered input or output data (including all asynchronous output). The error state is exited.

Remarks This event is enqueued when an error is detected and the Device's **State** transitions into the error state.

See Also "Error Handling" on page Intro-23, "Device Output Models" on page Intro-25.

[Goto Table 1-292](#)

OutputCompleteEvent

<<event>> `upos::events::OutputCompleteEvent`
OutputID : int32{read-only}

Description Notify the application that the queued output request associated with the `outputID` property has completed successfully.

Attributes This event contains the following attributes:

Attribute	Type	Description
<code>OutputID</code>	<code>int32</code>	The ID number of the asynchronous output request that is complete.

Remarks This event is enqueued after the request's data has been both sent and the Service has confirmation that it was processed by the device successfully.

See Also "Device Output Models" on page Intro-25

[Goto Table 1-293](#)

UPOS Ver1.16 RCSD Specification

StatusUpdateEvent

<<event>> `upos::events:: StatusUpdateEvent`
`Status` : `int32 {read-only}`

Description *Notifies the application that there is an operation status change or a status of the Graphic Display device.*

Attributes This event contains the following attribute:

<u>Attributes</u>	<u>Type</u>	<u>Description</u>
<i>Status</i>	<i>int32</i>	Indicates a change of operation status of graphic display device

Note that Release 1.3 added Power State Reporting with additional Power reporting StatusUpdateEvent values.

The Update Firmware capability added additional *Status* values for communicating the status/progress of an asynchronous update firmware process. See “StatusUpdateEvent” description on page 1-34.

<u>Value</u>	<u>Meaning</u>
--------------	----------------

GDSP_SUE_START_IMAGE_LOAD	It will be notified when image loading start.
---------------------------	---

GDSP_SUE_END_IMAGE_LOAD	It will be notified when image loading end.
-------------------------	---

GDSP_SUE_START_LOAD_WEBPAGE	Start loading the web page.
-----------------------------	-----------------------------

GDSP_SUE_FINISH_LOAD_WEBPAGE	It has finished loading the web page.
------------------------------	---------------------------------------

GDSP_SUE_CANCEL_LOAD_WEBPAGE	It has canceled loading the web page.
------------------------------	---------------------------------------

GDSP_SUE_START_PLAY_VIDEO	Start playing video.
---------------------------	----------------------

GDSP_SUE_STOP_PLAY_VIDEO	Stop playing video.
--------------------------	---------------------

Remarks Enqueued when the Graphic Display Device detects a power state change or a status change.

See Also “Events” on page Intro-19.

[Goto Table 1-294](#)

Relationship to other OMG specification and activities

Robotics Domain Task Force

Activities in Robotics Domain Task Force

The OMG Robotics Domain Task Force (Robotics DTF) fosters the integration of robotics systems from modular components through the adoption of OMG standards. It recommends the adoption and extends OMG technologies that apply to the specific domain of robotics systems where no current baseline specifications exist, such as MDA for Robotics. The object technology is not solely limited to software but is extended to real objects. It also collaborates with other organizations for standardization, such as the one for home information appliances, and makes an open effort to increase interoperability in the field of robotics.

(<https://www.omg.org/robotics/>)

RoIS Specification

Robotic Interaction Service Framework [RoIS] defines several functional components for robotic interaction services.

Definitions related to locations of entities in robotic services will be described with Robotic Localization Service[RLS]. Definitions of status of components in services will be described in conjunction with Robotic Technology Component [RTC], Finite State Machine Component for RTC [FSM4RTC] and Unified Component Model for Distributed Real-Time and Embedded Systems [UCM].

RoIS specification seeks that specify a RoIS framework, on top of which various service robot applications are developed.

Scope of RoIS specification

They are summarized in the following items.

- Interface between service application and Human Robot Interaction (HRI) engine
- Interface to obtain information from HRI Engine according to the timing of the service application's needs (Query)
- Interface to receive information from HRI Engine triggered by real time events (Event notification / subscription / cancellation)
- Interface for instructions to device control HRI Engine functions (Command)
- Definition of common messages for all HRI Engines

UPOS Ver1.16 RCSD Specification

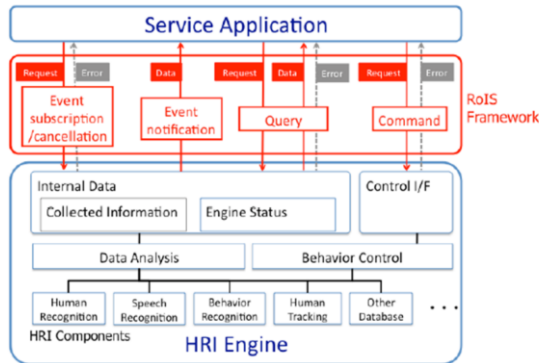


Fig.5: Example of RoIS Framework

Robot Service Ontology [RoSO] RFP

A new RFP of Robot Service Ontology[RoSO] currently being discussed in Robotics DTF are based on the concept of RoIS.

RoSO is aiming to define the specification (ontology) that clarifies the concept of a common vocabulary and / or a robot service in order to describe a service provided by a robot or exchange a description of a service provided by a service robot

Below is an example of HRI main component examples from this point of view.

Table K-1 – (From RoIS 1.2) Basic HRI Components

HRI Component Name	Description
system information	Provides the information of the system such as status of the system and position of the physical unit.
person detection	Detects number of people
person localization	Detects position of people
person identification	Identifies ID (name) of people
face detection	Detects number of human faces
face localization	Detects position of human faces
sound detection	Detects number of sound sources
sound localization	Detects position of sound sources
speech recognition	Recognizes person's speech
gesture recognition	Recognizes person's gesture
speech synthesis	Generates robot speech
reaction	Performs specified reaction
navigation	Moves to specified target location
follow	Follows a specified target object
move	Moves to specified distance or curve

UPOS Ver1.16 RCSD Specification

Interoperability between UPOS RCSD and RoIs

Relationship between UPOS RCSD and RoIs

OMG's Robotics standard provides a lower level control layer to manage Robot Device with finer granularity and higher accuracy to accommodate a wide range of industry applications.

On the other hand, the UPOS RCSD specification focuses on the functioning of robotic equipment within the retail store environment. In the UPOS RCSD specification robots are treated as peripheral equipment of the latest POS system. Therefore, the UPOS RCSD specification focuses on the definition of the interface between the POS and the robotic device.

RoIs is already existing as OMG standard and it defined a component frame service that was intended for robotic communication services with people.

Therefore, RoIs developed a general robot service framework, which is different from UPOS RCSD, but it is possible to describe the function of UPOS RCSD.

To confirm the compatibility and interoperability of the RCSD functions of RoIs and UPOS, both DTFs created and confirmed the function mapping table.

For this purpose, we use the general RoIs HRI component defined in the RoIs 1.2 specification.

UPOS RCSD Device and HRI Components Mapping Check Result

UPOS Device	RoIs HRI Component Name	Description
Capability(function) of each device	system information	Provides the information of the system such as status of the system and position of the physical unit.
Individual Recognition	person detection	Detects number of people
	person localization	Detects position of people
	person identification	Identifies ID (name) of people
	face detection	Detects number of human faces
	face localization	Detects position of human faces
	gesture recognition	Recognizes person's gesture
Sound & Voice Recognition	sound detection	Detects number of sound sources
	sound localization	Detects position of sound sources
	speech recognition	Recognizes person's speech
Speech Synthesis	speech synthesis	Generates robot speech
Gesture Control	reaction	Performs specified reaction
	navigation	Moves to specified target location
	follow	Follows a specified target object
	move	Moves to specified distance or curve
POS Power	Implementable as user defined Component	N/A
Lights		
Video Capture		
Sound Recorder		
Sound Player		
Device Monitor		
Graphic Display		

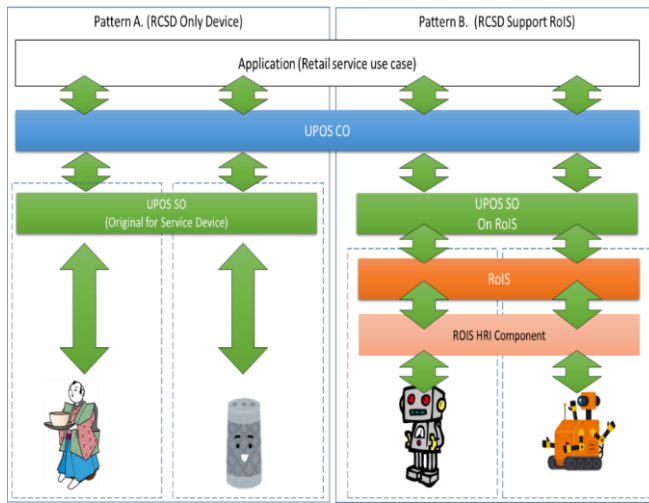
UPOS Ver1.16 RCSD Specification

The two teams continue to collaborate between the part of their separate RFP's and standards that will be established.

For that purpose, it is very necessary to understand the common vocabulary of the robot service and the needs of the ontology.

If each team's specification satisfies the above mapping table, it is confirmed that the standard can be maintained independently.

In addition, the figure below shows a typical scenario where RCSD and RoIS work independently or in conjunction.



UPOS Ver1.16 RCSD Specification

Document History

Version History

Ver	Date	Sections	Description of Change
1.0	2019-2-18		Initial Version – additions and updates to UPOS v1.15
1.1	2019-7-09		Revised for the issues and additions from the Review
1.2	2020-2-21		Issues, Updates are added version from the Review
1.3	2020-7-16		Issues, Updates are added version from the Review

Glossary

Term	Definition
EVRW	Electronic Value Reader Writer
CAT	Credit Authorization Terminal

UPOS Ver1.16 RCSD Specification

UPOS 1.16 RCSD Issues Table

Nwo.	Chapter	Section	Item	Company /Name	Issues	Editing Result	Comments	Status	Conclusion
Issue1 Table1	Preface	Doc No.	Reference Doc. No was incorrect.	SEIKO EPSON/ Tad Furuhata	This was found in the final checking	It was not retail/2019/04-01 but dtc/20-04-02.		Resolved	
Issue2 Table2	IPR Mode description	IPR Mode description	There was a typo.	SEIKO EPSON/ Tad Furuhata	There was a description "based on"	Corrected "based on"		Resolved	
Issue3 Table3	Document submitter.	Document submitter.	There were typos.	SEIKO EPSON/ Tad Furuhata	There were descriptions "Sumbitter" and "Supportes".	They were corrected "Submitter" and "Supporters".		Resolved	
Issue4 Table4	Table of Content	Table of Content	There was not a word of "Table of Content"	SEIKO EPSON/ Tad Furuhata	The word of Table of Content was missing in the Table of Content Section.	Added the word of "Table of Content" in the Table of Content section.		Resolved	
Issue5 Table5	Specification Overview	Specification Overview	There was a typo.	SEIKO EPSON/ Tad Furuhata	There was a description "Overview"	"Overview" was corrected to "Overview"		Resolved	
Issue6 Table6 Table13	21/Lights	Summary	SwitchOn	Diebold/ Dennis	switchOn() was added in 1.12 not 1.16	Correct the SwitchOn() method version number from 1.16 to 1.12.	Yes, it is a mistake as you pointed out.	Edited from 1.16 to 1.12	Resolved
Issue7 Table9 Table11 Table12 Table13	21/Lights	Property	CapFullColor property	Diebold/ Dennis	The full color concept needs more explanation in the General section. It is not clear what it means regarding HW device capabilities.	Since current Color property does have the function that described in FulColor property. Therefore, eliminate the FullColor and CapFullColor properties.	The description of the FullColor property was a mistake. At first, we were thinking of installing the FullColor property, but we realized that the Color property could satisfy the usage and canceled it. When the editor reflected in the UPOS specification, it was a mistake because it made it the former Japanese document base and reflected it. I have fixed the relevant parts.	FullColor and CapFullColor property will be eliminated	Resolved

UPOS Ver1.16 RCSD Specification

<p>Issue8 Table14</p>	<p>21/Lights</p>	<p>Method</p>	<p>switchOnMultiple method</p>	<p>Diebold/ Dennis</p>	<p>What is the reason for this method? Normally, the same can be achieved by several calls to switchOn(). The only reason, what comes in my mind is to achieve synchronous blinking. In that case it must state that way. Furthermore this method must be guarded by a capability as not all devices may be controlled in a way to blink synchronously.</p>	<p>For the needs to turn on the multiple light simultaneously this method was prepared. And this description was added in this method.</p>	<p>(7/12Yasumoto) Thank you for the good idea. Please let me use this. (6/6Dennis) OK, then the description should state it that way. "This method does the same as swithOn but in a synchronized way such that all lights are switched on/blinking synchronously" Yes, you can do the same by calling switchOn () multiple times. The reason for preparing this method is that if you want to light multiple light devices simultaneously, the switchOn () method takes time, and the user's needs may not be met, so there is a method that can be instructed to the device at one time. I prepared.</p>		<p>Resolved</p>
<p>Issue9</p>	<p>29/POS Power</p>	<p>Summary</p>	<p>ChargeTime</p>	<p>Diebold/ Dennis</p>	<p>The ChargeTime property should be accessible after enable as it needs communication to the device which is recommended after enabling only.</p>	<p>QuickChargeTime will not change the spec. this time, since It is not clear about this thinking way.</p>	<p>(6/6Dennis) If QuickChargeTime is wrong, we should not repeated this mistake... And maybe, we should correct this for the old properties too. Since the QuickChargeTime property, which is a property that existed before, was accessible before it was enabled, we decided to make this property accessible as well.</p>	<p>Charge time property accessible capability was same as current Quick Charge Time Property. Need to decide which way we should go. We will keep the original UPOS thinking way. ChargeTime can be accessible after open.</p>	<p>Not Adopted</p>
<p>Issue10 Table15 Table22</p>	<p>29/POS Power</p>		<p>TimeMode This property was eliminated</p>	<p>Diebold/ Dennis</p>	<p>The TimeMode property should be accessible after enable as it needs communication to the device</p>	<p>Eliminated the TimeMode property, instead added the BatteryCapacityRemaining Property.</p>	<p>Since the QuickChargeMode property, which is a property that existed</p>	<p>Time mode was accessible even before it is enabled. Since current spec was so.</p>	<p>Resolved</p>

UPOS Ver1.16 RCSD Specification

					which is recommended after enabling only.	BatteryCriticallyLowThresh old Property, BatteryLowThreshold Property. Each property handle the seconds.	before, was accessible even before it was enabled, we decided to make this property accessible as well.		
Issue11 Table18 Table19 Table20 Table21 Table23 Table24	29/POS Power	Property	TimeMode This property was eliminated	Diebold/Dennis	This property design is not extensible! I would prefer to have an enumeration with seconds and percent. If this is not acceptable, the name should be changed at least, e.g. "TimeInSeconds".		(6/6Dennis) Agree I agree with that opinion. We propose to add the property that handles seconds to the BatteryCapacityRemaining property, the BatteryCriticallyLowThreshold property, and the BatteryLowThreshold property by removing the TimeMode property.	Instead of TimeMode property we would like to propose those properties. That is to say, BatteryCapacityRemaining, BatteryCriticallyLowThreshold and BatteryLowThreshold properties.	Resolved
Issue12	29/POS Power		Syntax(access after open) Time mode is eliminated, therefore this discussion has no meaning.	Diebold/Dennis	Should be not open after enable, see Summary.	TimeMode Property's Syntax. Since it is removed from the POS/Power and we did not need the TimeMode related discussion.	The TimeMode property has been removed, so it is no longer necessary.	TimeMode property was deleted	Not Adopted
Issue13 Table48 Table51	39/Video Capture	Property	BarCodeEnabled property	Diebold/Dennis	There is no need to describe the Bar Code Scanner device function in the Video Capture device since it will be handled by Bar Code Scanner Service Object ad Hydra device.	There is no need to describe the Barcode Scanner function in the Video Capture device since it will be handled by Bar Code Scanner Service Object ad Hydra device.			This will be eliminated.
Issue14 Table49	39/Video Capture		CapIndividualRecognition property	Diebold/Dennis	There is no need to describe the Individual Recognition Device function in the Video Capture device since it will be handled by Bar Code Scanner Service Object as Hydra device.	There is no need to describe the Individual Recognition Device function in the Video Capture device since it will be handled by Bar Code Scanner Service Object ad Hydra device.			This will be eliminated.
Issue15 Table28 Table76	39/Video Capture		CapPhotograph Property =>This will be CapPhoto Property instead	Diebold/Dennis	Isn't "picture" a better name for the "photo" concept?	Consequently we thinks in here photo wording will be fit rather than picture.	Consequently we thinks in here photo wording will be fit rather than picture.		Resolved

UPOS Ver1.16 RCSD Specification

Issue16 Table38	39/Video Capture	Method	readFrame Method syntax readFrame(frameData: string):	Diebold/ Dennis	Hasn't this to be declared as "out" parameter as data is returned through it? Or, is it the file path to which the data will be stored? In that case the description has to be more explicit about that.	Consequently we eliminated the readFrame method.			Resolved
Issue17 Table39	39/Video Capture	Method	startVideRecording Method Remarks When the time specified in RecordingTime has elapsed, or'...' => We would like to use startVideo as the method.	Diebold/ Dennis	I recommend to issue an appropriate StatusUpdateEvent in this case.	We will use the StatusUpdateEvent to check this method's status.			Resolved
Issue18 Table40	39/Video Capture	Method	stopVideoRecording method =>stopVideo method	Diebold/ Dennis	I guess, it has to be stated here that method is processed synchronously. Means, it returns only when the recording has been stopped and video file has been written.	We will use the StatusUpdateEvent to check this method's status.			Resolved
Issue19	39/Video Capture	Method	takePhotograph Method =>takePhoto method	Diebold/ Dennis	?	We made the decision to use the takePhoto Method name even proposed to use take picture instead.			Resolved
Issue20 Table45	39/Video Capture	Method	take a photo	Diebold/ Dennis	take an image will be better	We made the decision to use the take a photo instead of taking an image.			Resolved
Issue21 Table45	39/Video Capture	Method	take a movie => take a vide has been chosen.	Diebold/ Dennis	take a video will be better	Instead of movie we will use the video in this device.			Resolved
Issue22 Table46	39/Video Capture	Model	Capture only mode => we will not use the word of capture in this Video Capture device behavior since this is camera device and capturing is very common word for camera and we decided not to use the wording of capture.	Diebold/ Dennis	Need to be formulated regarding those descriptions.	We will use the word of video instead of video capture.			Resolved
Issue23 Table46	39/Video Capture	Model	Photo shooting mode => we will use the photo wording instead of photo shooting	Diebold/ Dennis	Need to be formulated regarding those descriptions.	Instead of shooting we will use the recording.			Resolved
Issue24 Table46	39/Video Capture	Model	Photo shooting mode =>This will be a Photo mode.	Diebold/ Dennis	This should be image capturing	Instead of shooting we will use the recording.			Resolved

UPOS Ver1.16 RCSD Specification

Issue25 Table46	39/Video Capture	Model	Movie shooting mode => This will be Video mode.	Diebold/ Dennis	Need to be formulated regarding those descriptions.	We think it is good enough to use the Video wording in here.			Resolved
Issue26 Table46	39/Video Capture	Model	Movie shooting mode => This will be Video mode.	Diebold/ Dennis	This should be video capturing	We think it is good enough to use the Video wording in here.			Resolved
Issue27 Table47	39/Video Capture	Input Model	There is the description of Control	Diebold/ Dennis	OPOS should say device or device control.	In here eliminated the word of control.	In here eliminated the word of control.		Resolved
Issue28	39/Video Capture	Input Model	ends when the specified time elapses and recording to the specified file is completed.	Diebold/ Dennis	Why it is not complementary? This must be either issue the event on starts and stop or never issue the event at all.	Since this is input device model, we changed the Model description accordingly.	Input device model description should be used since this is the input device. OPOS-J changed the description accordingly.		Resolved
Issue29	39/Video Capture	Input Model	When an application calls the stopVideoRecording method to end recording, DataEvent event will not occur. "	Diebold/ Dennis	Need to remove	SUE will notify the end of status.	Added the SUE for this model.		Resolved
Issue30	39/Video Capture	Input Model	Also, by activating the FaceCatchEnabled property, face recognition is started, and even when a face is recognized, a DataEvent event is generated.	Diebold/ Dennis	FaceCatchEnabled Property has been eliminated since this is the function of individual recognition and it is not a function of Video Capture Device.				Resolved
Issue31	39/Video Capture	Input Model	To distinguish between Recording Completed to File by Recording and DataEvent event of Face Recognition, refer to the DataEventType property.	Diebold/ Dennis	FaceCatchEnabled Property has been eliminated since this is the function of individual recognition and it is not a function of Video Capture Device.				Resolved
Issue32 Table47	39/Video Capture	Input Model	The control sets VCP_ET_VIDEO when recording to the file by recording is completed, and sets VCP_ET_FACECATCH to the DataEventType property when recognizing the face. "	Diebold/ Dennis	FaceCatchEnabled Property has been eliminated since this is the function of individual recognition and it is not a function of Video Capture Device.				Resolved
Issue33 Table26	39/Video Capture	Input Model	If the DataEventEnabled property is true, the queued DataEvent is notified to the application. Just before triggering this event, the control copies the data to the property and sets the DataEventEnabled property to false to prevent further data events	Diebold/ Dennis	Please check the edited model description.	Edited the Model description completely.			Resolved

UPOS Ver1.16 RCSD Specification

			firing. This allows the control to queue subsequent input data while the application is processing the current input and processing the related properties. When the application finishes processing the current input data and is ready for the next data processing, setting the DataEventEnabled property to true will notify the Data Event again.						
Issue34 Table25 Table47	39/Video Capture	Input Model	Control	Diebold/Dennis	UPOS device spec should not describe the "control".	Eliminated the word of control.	Eliminated the word of control.		Resolved
Issue35 Table35	39/Video Capture	Bar Code Scan	Video capture	Diebold/Dennis	Bar Code Scan function was eliminated completely. Since it will be used as hydra device and all of the Bar Code function description has been eliminated.		Bar Code will be used hydra device therefore there is no description in this chapter.		Resolved
Issue36 Table35	39/Video Capture	Bar Code Scan	When reading data from the bar code, the DataEvent event is queued in the scanner service object.	Diebold/Dennis	Bar Code Scan function was eliminated completely. Since it will be used as hydra device and all of the Bar Code function description has been eliminated.	Changed the relationship between Bar Code Scan device and Individual recognition device completely.	Bar Code will be used hydra device therefore there is no description in this chapter.		Resolved
Issue37 Table35 Table74	39/Video Capture	Bar Code Scan	Scanned data is stored in the ScanData property. If the application sets the DecodeData property to true, the data is decoded to ScanDataLabel and ScanDataType.	Diebold/Dennis	Bar Code Scan function was eliminated completely. Since it will be used as hydra device and all of the Bar Code function description has been eliminated.	Changed the relationship between Bar Code Scan device and Individual recognition device completely.	Bar Code will be used hydra device therefore there is no description in this chapter.		Resolved
Issue38 Table75	39/Video Capture	IndividualRecognition		Diebold/Dennis	Individual Recognition device function has been eliminated completely. Since that will be handled by individual Recognition device as hydra connection.	Changed the relationship between video capture device and Individual recognition device completely.	Individual Recognition will be used as hydra device thefore there is no description in this chapter.		Resolved
Issue39 Table75	39/Video Capture	IndividualRecognition	The detected data is stored in the IndividualRecognitionInformation and IndividualIDs of Individual Recognition Device properties.	Diebold/Dennis	Individual Recognition device function has been eliminated completely. Since that will be handled by individual Recognition device as hydra connection.	Changed the relationship between video capture device and Individual recognition device completely.	Individual Recognition will be used as hydra device thefore there is no description in this chapter.		Resolved

UPOS Ver1.16 RCSD Specification

Issue40 Table75	39/Video Capture	IndividualRecognition	If the property is true, it indicates that you can use a VideoCapture device to read barcodes. If you want to read the barcode, you can use it by opening the Scanner device separately.	Diebold/Dennis	Individual Recognition device function has been eliminated completely. Since that will be handled by individual Recognition device as hydra connection.	Changed the relationship between video capture device and Individual recognition device completely.	Individual Recognition will be used as hydra device therefore there is no description in this chapter.		Resolved
Issue41 Table30	39/Video Capture	Property	CameraGain Property => Gain Property	Diebold/Dennis	What happens with the property value if CameraAutoGain is true? This should be stated here!	Added the explanation about the AutoCameraGain Gain property into the CameraGain property.	Added the explanation both CameraGain Property and CameraAutoGain property.		Resolved
Issue42 Table30 Table94 Table95 Table96 Table97 Table98	39/Video Capture	Property	CameraHorizontalFlip Property =>Horizontal Flip Property	Diebold/Dennis	Flipping, horizontally and vertically, is not described in the model of VideoCapture device. I'm not an camera expert. But I'm wondering what happens, when bot properties CameraHorizontalFlip and CameraVerticalFlip are both true...	Added the additional explanation into the CameraHorizontalFlip property section.	Added the additional explanation into the HorizontalFlip and VerticalFlip. Also decided to eliminate the words of Camera.		Resolved
Issue43 Table30 Table112	39/Video Capture	Property	CameraVerticalFlip Property => Vertically Flip	Diebold/Dennis	Same as CameraHorizontalFlip Property	Added the additional explanation into the CameraHorizontalFlip property section.	Added the additional explanation into the HorizontalFlip and VerticalFlip.		Resolved
Issue44 Table29 Table52 Table58	39/Video Capture	Property	CapCameraAutoExposition Property =>CapAutoExposure If true, can change the auto exposition of camera. If false, cannot change the auto exposition of camera.	Diebold/Dennis	English, rephrasing needed: "If true, the camera supports auto exposition to be controlled by the property CamerAutoExposition" è native speaker	Changed the remarks description from previous one to new one.	Improve the description regarding the VideoCapture Property.		Resolved
Issue45 Table29 Table53 Table59	39/Video Capture	Property	CapCameraAutoFocus Property =CapAutoFocus Property If true, can change the auto focus of camera. If false, cannot change the auto focus of camera.	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec. If you have any other better expression please let me know.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved
Issue46 Table29 Table54 Table60	39/Video Capture	Property	CapCameraAutoGain Property =>CapAutoGain Property If true, automatic gain change of the camera is possible. If false, automatic	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved

UPOS Ver1.16 RCSD Specification

			gain change of camera is not possible.		If you have any other better expression please let me know.				
Issue47 Table29 Table55 Table61	39/Video Capture	Property	CapCameraAutoWhiteBalance Property =>CapAutoWhiteBalance Property If true, auto white balance of camera is possible. If false, auto white balance of camera is not possible.	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec. If you have any other better expression please let me know.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved
Issue48 Table29 Table56 Table62	39/Video Capture	Property	CapCameraBrightness Property CapBrightness Property If true, the brightness of camera can be changed. If false, the brightness of the camera cannot be changed.	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec. If you have any other better expression please let me know.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved
Issue49 Table29 Table63	39/Video Capture	Property	CapCameraContrast Property =>CapContrast Property If true... If false,...	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec. If you have any other better expression please let me know.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved
Issue50 Table64	39/Video Capture	Property	CapCameraExposure Property =>CapExposure Property If true... If false...	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec. If you have any other better expression please let me know.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved
Issue51 Table29 Table65	39/Video Capture	Property	CapCameraGain Property =>CapGain Property If true... If false...	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec. If you have any other better expression please let me know.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved

UPOS Ver1.16 RCSD Specification

Issue52 Table29 Table66	39/Video Capture	Property	CapCameraHorizontalFlip Property =>CapHorizontalFlip Property If true... If false...	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec. If you have any other better expression please let me know.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved
Issue53 Table29 Table67	39/Video Capture	Property	CapCameraHue Property => CapHue Property If true... If false...	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec. If you have any other better expression please let me know.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved
Issue54 Table29 Table83 Table110	39/Video Capture	Property	CapCameraSaturation Property =>CapSaturation Property If true... If false...	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec. If you have any other better expression please let me know.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved
Issue55 Table29 Table85	39/Video Capture	Property	CapCameraVerticalFlip Property =>CapVerticalFlip Property If true... If false...	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec. If you have any other better expression please let me know.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved
Issue56 Table31 Table68 Table86 Table99	39/Video Capture	Property	CapCapture Property =>CapVideo If true... If false...	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec. If you have any other better expression please let me know.	Changed the remarks description from previous one to new one. Also decided not use the word of Capture.	Please refer to the revised description.		Resolved
Issue57 Table31 Table69 Table77 Table87	39/Video Capture	Property	CapCaptureColorSpace Property =>CapVideoColorSpace Property If true, ... If false, ...	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved

UPOS Ver1.16 RCSD Specification

					If you have any other better expression please let me know.				
Issue58 Table31 Table71 Table78 Table88	39/Video Capture	Property	CapCaptureFrameRate Property =>CapVideoFrameRate Property If true, ... If also, ...	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec. If you have any other better expression please let me know.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved
Issue59 Table31 Table79 Table90	39/Video Capture	Property	CapCaptureResolution Property =>CapVideoResolution Property If true, ... If false, ...	Diebold/Dennis	English, rephrasing needed. à native speaker. => We think just using the same way as current UPOS spec. If you have any other better expression please let me know.	Changed the remarks description from previous one to new one.	Please refer to the revised description.		Resolved
Issue60 Table31	39/Video Capture	Property	CapCaptureResolutionList Property VideoResolutionList Property ."320 x 240, 640 x 480, 640 x 360".	Diebold/Dennis	Are the spaces required or optional? Needs to be defined clearly!	Eliminated the spaces in the resolution description. Also changed the property name the same as current UPOS way.	Please refer to the revised description.		Resolved
Issue61 Table32	39/Video Capture	Property	CapIndividualRecognition Property What is an "individual recognition"? Needs to be explained in the model section. =>This was eliminated since this will be supported as different device.	Diebold/Dennis	If the property is true, it indicates that you can use the VideoCapture device to create an IndividualRecognition device. It can be used by opening IndividualRecognition device separately. This was added to the specification. =>It is not added in Model section yet.	Individual Recognition device will be use as hydra device therefore, there is no description in this chapter.	Please refer to the revised description.		Resolved
Issue62 Table76	39/Video Capture	Property	CapPhotograph Property =>CapPhoto Property	Diebold/Dennis	Isn't "picture" a better name for the "photograph" concept? =>In the discussions by Japanese members, it was the opinion that photographs, rather than pictures, would be better for expressing photography. It is an opinion by non-native language members. Is picture	We decided to use the photograph not a picture. =>Consequently we will use photo instead of photograph.	OPOS-J prefer to use the photograph not a picture. OPOS-J prefer to use the photo.		Resolved

UPOS Ver1.16 RCSD Specification

					appropriate? =>English native speaker needed here!				
Issue63 Table33	39/Video Capture	Property	CapPhotographType Property =>CapPhotoType Property If true, ... If false, ...	Diebold/ Dennis	English, rephrasing needed. à native speaker.	Changed the remarks description from previous one to new one. Changed the property name the same as current UPOS way.	Please refer to the revised description.		Resolved
Issue64 Table33 Table81 Table108	39/Video Capture	Property	CapPhotographTypeList Property => PhotoTypeList property "BMP, JPEG"	Diebold/ Dennis	Are whitespace optional or mandatory? Needs to be clearly stated here.	Basically there is no space this is our thinking way.	Basically we will eliminate the white space. We think it is OK. Also we changed the property name based on the current UPOS way.		Resolved
Issue65 Table120	39/Video Capture	Property	CapVideoRecording Property => CapVideo Property If true, ... If false, ... Description of "movie" is OK or not.	Diebold/ Dennis	English, rephrasing needed. à native speaker. Video and movie are there, should be one description will be needed.	Changed the remarks description from previous one to new one. Changed the property name the same as current UPOS way.	Please refer to the revised description.		Resolved
Issue66 Table91	39/Video Capture	Property	CapVideoRecordingResolutionList Property =>VideoResolutionList Property Resolution is indicated by "Horizontal x Vertical" format.	Diebold/ Dennis	Are whitespaces required or optional? Needs to be stated clearly!	Basically there is no space this is our thinking way. And changed the property name the same as current UPOS way.	We elinameted the white space. We think now it is OK.		Resolved
Issue67 Table92	39/Video Capture	Property	CapVideoRecordingType Property =>CapVideoType Property If true,... If false,...	Diebold/ Dennis	English, rephrasing needed. à native speaker.	Changed the remarks description from previous one to new one.	Please refer to the revised description. Also changed the property name the same as current UPOS way.		Resolved
Issue68 Table93	39/Video Capture	Property	CapVideoRecordingTypeList Property =>VideoTypeList Property For example, when AVI_IYUV, AVI_MJPG is supported, it is the following. "AVI_IYUV, AVI_MJPG"	Diebold/ Dennis	Where these values comes from? Is it standardized? If yes, a reference is needed. If not, a naming schema should be described at least.	Yes it is a very common description who is handling the video images. Also changed the property name the same as current UPOS way.	AVI, IYUV, MJPG are very popular description in the Video technology.		Resolved
Issue69 Table106	39/Video Capture	Property	IndividualRecognitionEnabled Property => This property will be eliminated. If true,... If false, ...	Diebold/ Dennis	More explanation needed here about associated Individual recognition device.	Individual Recognition device will be used as hydra device therefore, there is no description in this chapter.	Eliminated the Individual Recognition device description in this chapter.		Resolved

UPOS Ver1.16 RCSD Specification

Issue70 Table103 Table104	39/Video Capture	Property	PhotographResolution Property =>PhotoResolution Property This property is referenced only when VCP_VCM_PHOTO is set in VideoCaptureMode property.	Diebold/Dennis	Referenced should be processed.	Edited the PhotographResolution Property. Also changed the name of this property to PhotoResolution.	Please refer to the PhotographResolution Property.		Resolved
Issue71 Table107	39/Video Capture	Property	PhotographType Property =>PhotoType Property	Diebold/Dennis	Referenced should be processed. There are duplicated remarks and there are two referenced description.	Edited. Also changed the property name to PhotoType	Please refer to the PhotographType Property.		Resolved
Issue72 Table113	39/Video Capture	Property	VideoCaptureMode Property acquired movie shooting	Diebold/Dennis	I'm a little bit confused: What is the difference between this capture mode and the other two modes? Ask differently: what is captures if not an image or a video? Should be stated in General section more clearly. Captured will be OK or not Movie shooting is a right expression or not.	We just changed this only two modes. That is to say video mode and photo mode. Others can be done by another device as hydra device.	Please refer to the model section that is explaining video mode and phot mode.		Resolved
Issue73 Table50 Table116	39/Video Capture	Property	VideoRecordingFrameRate Property movie taken Refereed =>VideoFrameRate Property, therefore this will handle only the video mode's frame rate.	Diebold/Dennis	Is movie taken is a good expression? Is referred is a good expression?	Made the decision to use the take a video and photo.	We will use the word of video for this property.		Not Adopted
Issue74 Table123	39/Video Capture	Method	readFrame Method syntax readFrame(frameData: out string): =>We eliminated the readFrame method.	Diebold/Dennis	out parameter was already added.	We agreed to use the out parameter into the readFrame method.	The readFrame method is eliminated.		Not Adopted
Issue75 Table36 Table109 Table124	39/Video Capture	Method	startVideoRecording method =>startVideo method recordingTime: Specify the time for recording in seconds. If FOREVER (-1) is specified, recording will continue until the stopVideoRecording method is called. When the time specified in RecordingTime has elapsed,	Diebold/Dennis	1) After reflecting the mode in Sound Recorder device I have the following model concern: This makes a mix of asynchronous and synchronous behavior. I'm not agree with this model and would suggest to go make it completely asynchrony! This means, all calls to	We edited this not the Input device model but a StatusUpdateEvent driven model based on the suggestion. Also added the property to handle the precise remaining recording time handling added the RemainingRecordingTimeIn sec property.	Please refer to the revised model and method description about this.		Resolved

UPOS Ver1.16 RCSD Specification

					<p>startRecording are asynchronous and will always result in a Start and Stop status update event. If recordingTime is FOREVER then stopRecording MUST be called. Otherwise, stopRecording may be called or is automatically stopped when the recordingTime elapsed. Start and Stop recording status update events are issued always accordingly.</p> <p>2) I recommend to issue an appropriate StatusUpdateEvent in this case. The SUE should be mentioned here, if added!</p>				
<p>Issue76 Table125</p>	39/Video Capture	Method	<p>stopVideoRecording method =>stopVideo method This method handles synchronously.</p>	Diebold/Dennis	<p>I guess, it has to be stated here that method is processed synchronously. Means, it returns only when the recording has been stopped and video file has been written. =>Thank you for your valuable opinion. I added a supplement. =>"Methods are processed synchronously." OK. "processed" is better than "handled", I guess.</p>	<p>We edited this not the Input device model but a StatusUpdateEvent driven model based on the suggestion.</p>	<p>Please refer to the stopVideo method description.</p>		Resolved
<p>Issue77 Table41 Table126</p>	39/Video Capture	Method	<p>takePhotograph Method =>takePhoto method.</p>	Diebold/Dennis	<p>You can take a picture by calling this method. After all, is this method name suitable for takePicture? =>I guess, "takePicture" is more suitable here. Native speaker required!</p>	<p>We edited this not the Input device model but a StatusUpdateEvent driven model based on the suggestion.</p>	<p>Please refer to the takePhoto method description.</p>		Resolved

UPOS Ver1.16 RCSD Specification

<p>Issue78 Table42 Table47 Table127</p>	<p>39/Video Capture</p>	<p>Event</p>	<p>DataEvent Before this event is delivered, the Video Capture movie image is placed into readFrame. This event is to be used only for those types of vendor specific functions that are not otherwise described. Use of this event may restrict the application program programform being used with other vendor's devices which may not have any knowledge of the Service's need for this event.</p>	<p>Diebold/Dennis</p>	<p>1) Model concern: I would not issue an event when readFrame method call returns as it is a synchronous call. DataEvents are typically issues for asynchronous calls to indicated the availability of a data in a property. 2)This is very vague. A more detailed description is required. 3)This is not the right model! I disagree strongly! For vendor specific events we have DirectIOEvent!</p>	<p>We edited this not the Input device model but a StatusUpdateEvent driven model based on the suggestion.</p>	<p>Please refer to the Model description regarding this device model about the Status notification.</p>		<p>Resolved</p>
<p>Issue79 Table129</p>	<p>39/Video Capture</p>	<p>Event</p>	<p>StatusUpdateEvent Value & Meaning VCP_SUE_START_VIDEO_RECORDING =>VCP_SUE_START_VIDEO It will be notified when video recording starts. VCP_SUE_STOP_VIDEO_RECORDING =>VCP_SUE_STOP_VIDEO It will be notified when video recording stop.</p>	<p>Diebold/Dennis</p>	<p>Mode concern: Generally, I would either make start/stopVideoRecording asynchronous and issue events. Or, make them synchronous and not issue events. Not both or mixed.</p>	<p>We changed the description regarding the StatusUpdateEvent.</p>	<p>We changed the description regarding the StatusUpdateEvent. Please refer to the revised StatusUpdateEvent description.</p>		<p>Resolved</p>
<p>Issue80</p>	<p>40 / Individual Recognition</p>	<p>Name of Device</p>	<p>Individual or Object which is better?</p>	<p>Diebold/Dennis</p>	<p>English, rephrasing needed. à native speaker.</p>	<p>We decided to use the Individual recognition instead of object recognition.</p>	<p>We decided to use the Individual recognition instead of object recognition.</p>		<p>Not Adopted</p>
<p>Issue81 Table133 Table136</p>	<p>40 / Individual Recognition</p>	<p>Property</p>	<p>IndividualRecognitionFilter Property supported functions are defined by the device.</p>	<p>Diebold/Dennis</p>	<p>Where they are defined? I would have expected a Capability for that. Or, is it such complex that it is required to be described device specific in the device documentation?</p>	<p>Added the individual Recognition Filter example in the specification.</p>	<p>Please refer to the individual Recognition Filter and its related documentation.</p>		<p>Resolved</p>
<p>Issue82 Table132 Table149</p>	<p>41/ Sound Recorder</p>	<p>Model</p>	<p>Model •"The control will generate a DataEvent when the recording started by the startRecording method ends when the specified time elapses and the recording to the specified file is completed.</p>	<p>Diebold/Dennis</p>	<p>1) Model concern: Same as for Video Capture device: either make both start and stop methods asynchronous and issue events. Or, make both synchronous and do not issue events. Not both or</p>	<p>Since this is the device Input Model and we edited the model section based on the Input device model' common phrase. Added the SoundData Property and we think it</p>	<p>Edited the Sound Recorder model description please refer to the revised model description about this device.</p>		<p>Resolved</p>

UPOS Ver1.16 RCSD Specification

			<p>•When an application calls the stopRecording method to end recording, DataEvent will not occur."</p> <p>•If the AutoDisable property is true, then the device automatically disables itself when a DataEvent is enqueued.</p> <p>•An enqueued DataEvent can be delivered to the application when the DataEventEnabled property is true and other event delivery requirements are met. Just before delivering this event, data is copied into corresponding properties, and further data events are disabled by setting DataEventEnabled to false. This causes subsequent input data to be enqueued while the application processes the current input and associated properties. When the application has finished processing the current input and is ready for more data, it reenables events by setting DataEventEnabled to true.</p>		<p>mixed! See detailed explanation at startRecording. 2) Model concern: This mode does not fit as there are no properties which get filled when a DataEvent is issued. However, this is the main design principle for DataEvents in UnifiedPOS 1! I would replace it by Status Update Events.</p>	<p>make sense to behave as Device Input Model.</p>			
<p>Issue83 Table155</p>	41/ Sound Recorder	Property	<p>CapChannelList Property Contains the comma-delimited list of channel that is supported by the device. For example, if the device only supports 1ch and 2ch and 4ch, then this property should be set to "1,2,4".</p>	Diebold/Dennis	<p>1) Missing the channels of s 2) Not clear the meaning of Ch.</p>	<p>Change the description of channel from Ch. to channel.</p>	<p>Change the description of channel from Ch. to channel.</p>		Resolved
<p>Issue84 Table158</p>	41/ Sound Recorder	Property	<p>CapSamplingRateList Property For example, if the device only supports 44.1KHz and 48KHz and 96KHz, then this property should be set to "44100,48000,96000".</p>	Diebold/Dennis	<p>It should be explicitly stated that the measurement of all values in the list is kHz.</p>	<p>Edited the frequency description from KHz to kHz.</p>	<p>Edited the frequency description from KHz to kHz.</p>		Resolved
<p>Issue85 Table161</p>	41/ Sound Recorder	Property	<p>CapSoundTypeList Property For example, if the device only supports WAV and OGG, then this property should be set to "WAV,OGG".</p>	Diebold/Dennis	<p>What's about lower case, upper case? I would recommend to make it case insensitive.</p>	<p>Basically WAV, OGG should be upper case.</p>	<p>Basically WAV, OGG should be upper case.</p>		Resolved
<p>Issue86 Table143 Table157</p>	41/ Sound Recorder	Method	<p>startRecording Method fileName Specify the file name of the image to</p>	Diebold/Dennis	<p>1) Wrong description. Copy&Paste error, I guess. 2) Model concern: This makes</p>	<p>Edited all of the copy and paste mistake and revised. In addition added the</p>	<p>Edited all of the copy and paste mistake and revised. In addition added the</p>		Resolved

UPOS Ver1.16 RCSD Specification

Table160 Table164			be loaded. recordingTime Specify the time for recording in seconds. If OPOS_FOREVER (-1) is specified, recording will continue until you call the stopRecording method.		a mix of asynchronous and synchronous behavior. I'm not agree with this model and would suggest to go make it completely asynchronous! This means, all calls to startRecording are asynchronous and will always result in a start and stop status update event. If recordingTime is FOREVER them stopRecording MUST be called. Otherwise, stopRecording may be called or is automatically stopped when the recordingTime elapsed. Start and Stop recording status update events are issued always accordingly. [Same as for VideoCapture start/stopRecording methods]	description regarding the StatusUpdateEvent firing.	description regarding the StatusUpdateEvent firing.		
Issue87 Table165	41/ Sound Recorder	Method	stopRecording Method Finish the recording and complete the recording of the audio file.	Diebold/ Dennis	In my proposed model (see startRecording) this method is always asynchronous and will always result in Stop event issuing...	stopRecording method is a synchronously behave and when this method is invoked StatusUpdateEvent may fire in accordance with the device state change.	Please refer to the stopRecording method and Model description about this device.		Resolved
Issue88	41/ Sound Recorder	Event	ErrorEvent	Diebold/ Dennis	If we make startRecording asynchrony, we need a proper error reporting through error events!	This method handles synchronously. We edited the model section very much.	Please refer to the edited Sound Recorder Model section.		Resolved
Issue89	42 / Voice Recognition	Function Name	HearingDataPattern:	Diebold/ Dennis	Is "Recognized" maybe the better word part here? native speaker Change for all occurrences!	We prefer to use the Hearing instead of Recognizing.	We prefer to use the Hearing instead of Recognizing.		Not Adopted
Issue90	42 / Voice Recognition	Property	CapLanguage Property CapLanguage: boolean {read-only, access after open}	Diebold/ Dennis	access after open & claim, enable	Changed as access after open-claim-enable	Here should be accessible after open.		Resolved
Issue91 Table170 Table174	42 / Voice Recognition	Property	HearingDataPattern Property HearingDataPattern: string {read-only, access after open}	Diebold/ Dennis	access after open & claim, enable	Changed as access after open-claim-enable	We changed this accessible after open-claim-enable.		Resolved

UPOS Ver1.16 RCSD Specification

Issue92 Table175	42 / Voice Recognition	Property	HearingDataWord Property HearingDataWord: string {read-only, access after open}	Diebold/Dennis	access after open & claim, enable	Changed as access after open-claim-enable	We changed this accessible after open-claim-enable.		Resolved
Issue93 Table176	42 / Voice Recognition	Property	HearingDataWordList Property HearingDataWordList: string {read-only, access after open} For example, in the startHearingSentence method, set candidates as follows, Word list: "Item: coffee; tea, number: one: two"	Diebold/Dennis	1) access after open & claim, enable 2) Are whitespaces optional or mandatory? Specify clearly.	Changed as access after open-claim-enable	We changed this accessible after open-claim-enable.		Resolved
Issue94 Table177	42 / Voice Recognition	Property	HearingResult Property HearingStatus: int32 {read-only, access after open} Value & Meaning TTS_HRESULT_YESNO_YES Voice recognition result of Finish running voice recognition. method. Also, Device got an answer that is classified as YES. The recognition content is set in the Finish running voice recognition property. TTS_HRESULT_YESNO_NO Voice recognition result of Finish running voice recognition. method. Also, Device got an answer that is classified as NO. The recognition content is set in the HearingDataWord property.	Diebold/Dennis	1) access after open & claim, enable 2) TTS_HRESULT_YESNO_YES Explanation is unclear. 3)TTS_HRESULT_YESNO_NO Explanation is unclear.	Edit this property based on the suggestion.	Please refer to the edited description of HearingResult property.		Resolved
Issue95 Table179	42 / Voice Recognition	Method	startHearingFree Method Remarks	Diebold/Dennis	The remarks must specify that the property HearingStatus is set accordingly BEFORE it returns.	Added the description in remarks.	Please refer to the edited startHearingFree Method description.		Resolved
Issue96 Table180	42 / Voice Recognition	Method	startHearingSentence Method Parameter pattern Remarks	Diebold/Dennis	1) I would recommend to define a regular expression here which matches it. This would also be easy to implement on all sides using a common RegEx implementation. 2) The remarks must specify that the property HearingStatus is set	Added the description in remarks.	Please refer to the edited startHearingSentence Method description.		Resolved

UPOS Ver1.16 RCSD Specification

					accordingly BEFORE it returns.				
Issue97 Table181	42 / Voice Recognition	Method	startHearingWord Method 1) Parameter wordList Example: "word 1, word 2, word 3" 2) Remarks	Diebold/Dennis	1)Are whitespaces optional or mandatory? Specify clearly. 2)The remarks must specify that the property HearingStatus is set accordingly BEFORE it returns.	Eliminated the white spaces. Added the description in remarks.	Please refer to the edited startHearingWord Method description.		Resolved
Issue98 Table182	42 / Voice Recognition	Method	startHearingYesNo Method Remarks	Diebold/Dennis	The remarks must specify that the property HearingStatus is set accordingly BEFORE it returns.	Added the description in remarks.	Please refer to the edited startHearingYesNo Method description.		Resolved
Issue99 Table183	42 / Voice Recognition	Method	stopHearing Method Remarks	Diebold/Dennis	1) This methods should not return before "hearing" /recognition is finished. In that sense it is synchronous ... 2) The remarks must specify that the property HearingStatus is set accordingly BEFORE it returns. 3) What's about events. They have not been defined here and are lacking!	Add the remarks. Also added the SUE description	Please refer to the edited stopHearing Method description.		Resolved
Issue100 Table191 Table199	43 / Sound Player	Model	Model 1) " 1. Buffers the request in program memory, for delivery to the Physical Device as soon as the Physical Device can receive and process it. 2. Sets the OutputID property to a unique integer identifier for this request. 3. Returns as soon as possible." 2) Applications need to support "hard total" services as audio files played with the startSound method must be placed in the area managed by the "hard total" service.	Diebold/Dennis	1) Formatting needs rework. 2) Why this reference to the Hard Total service. I think, it is not required. It is sufficient to assume the files are on the local hard disk. They even may be remote available.	Revised based on the suggestion. Also, changed regarding the hard total, from must to use to may use.	Please refer to the Sound Player device model description. And description of this method.		Resolved

UPOS Ver1.16 RCSD Specification

Issue101 Table197	43 / Sound Player	Property	Volume Property Volume : int32 {read-write, access after open, claim}	Diebold/Dennis	after open claim enabled.	Changed the behavior accessible after open-claim-enabled.	Changed the behavior accessible after open-claim-enabled.		Resolved
Issue102 Table198	43 / Sound Player	Method	playSound Method Remarks Audio files must be located in the area managed by "Hard Total" service.	Diebold/Dennis	Why this reference to the Hard Total service? I think, it is not required. It is sufficient to assume the files are on the local hard disk. They even may be remote available.	Reference to Hard Totals changed from must to may use.	Please refer to the description of model and method about this.		Resolved
Issue103	44/ Speech Synthesis	Name of Device	General Information Capabilities Convert text to speech and speak.	Diebold/Dennis	English: Needs rephrasing. à native speaker	Description regarding the speech and speak has been changed to speech and read it aloud.	Please refer to the revised description about the Speech Synthesis capability description.		Not Adopted
Issue104 Table207	44/ Speech Synthesis	Model	Model The application calls a speak method or speakImmediate method to speech. => Added description The speak method acts to start speaking the words specified by text, while the speakImmediate method ends immediately previous speak method, and starts speaking the word specified by text asynchronously and immediately.	Diebold/Dennis	Those explanation was already added. The speak method acts to start speaking the words specified by text, while the speakImmediate method ends immediately previous speak method, and starts speaking the word specified by text asynchronously and immediately.	Made the improvement regarding the speakMethod and added the description regarding the OutputCompleteEvent.	Added the model description based on the suggestion.		Resolved
Issue105 Table207	44/ Speech Synthesis	Property	OutputIDList Property	Diebold/Dennis	Normally, we do not maintain such lists in the device. It's the duty of the application...	We think we need this, since we can indicate the capability how many and what kinds of utterance can be done by this targeted device.	Also please refer to the device model behavior description.		Resolved
Issue106 Table208	44/ Speech Synthesis	Method	speak Method 1) Tags without reset are specified in the form of "\ tag = value \". For example, when specifying Text as follows, "Hello \ pause = 1000 \ \ pitch = 150 \ \ It's nice weather today \ \ reset \". "Hello" speaks according to the original setting. Then wait for 1000 milliseconds. "Today" speaks Pitch at 150%. "Nice weather," I will speak according to the original settings. 2) This method is executed asynchronously.	Diebold/Dennis	1) I would like to introduce another meta syntax for speak control: "[tag=value]" This is more common to programmers than backslashes. backslashes are also problematic in Unicode programming environments (meta signs there). 2)What's about the OutputID. It should be stated here that the property OutputID is set accordingly.	We would like to propose to use the table for this description. Please take a look at the revised description.	We would like to keep current function of this method. On the other hand, added the method behavior of OutputCompleteEvent.		Resolved

UPOS Ver1.16 RCSD Specification

Issue107 Table209	44/ Speech Synthesis	Method	speakImmediate Method This method is executed asynchronously.	Diebold/Dennis	I would make this method synchronous only, see my main comment about that in the general section.	This will be the same as speak method.	This will be the same as speak method.		Resolved
Issue108 Table210	44/ Speech Synthesis	Method	stopCurrentSpeaking Method Remarks	Diebold/Dennis	What is about the OutputCompleteEvent. Is it issued before the method returns? It should!	Can check the status by using the StatusUpdateEvent. Please refer to the model section.	Can check the status by using the StatusUpdateEvent, since this method behave asynchronously. Please refer to the model section.		Resolved
Issue109 Table207	44/ Speech Synthesis	Method	stopSpeaking Method Remarks	Diebold/Dennis	What is about the OutputCompleteEvent. Is it issued before the method returns? It should!	Can check the status by using the StatusUpdateEvent. Please refer to the model section.	Can check the status by using the StatusUpdateEvent, since this method behave asynchronously. Please refer to the model section.		Resolved
Issue110 Table220	45 / Gesture Control	General Info	General Information 1) The Gesture Control programmatic name is "Gesture Control".	Diebold/Dennis	We are not talking about Controls in the UnifiedPOS spec, use "device" instead (see Scanner, e.g.).	Eliminated the control from the description.	Eliminated the control from the description.		Resolved
Issue111 Table220	45 / Gesture Control	Capabilities	Capabilities 1) The Gesture Control has the following capability: 2) •It controls the operation of various joints.	Diebold/Dennis	1) We are not talking about Controls in the UnifiedPOS spec, use "device" instead (see Scanner, e.g.). 2) What are "joints". Is that the right phrasing? Maybe, "move" is better here? à native speaker	Changed the control to device control. Changed the description from the operation of various joints to the behavior of various joint components and parts.	Changed the control to device control. Changed the description from the operation of various joints to the behavior of various joint components and parts.		Resolved
Issue112 Table221	45 / Gesture Control	Model	Pose / Motion Since the created pause and motion files are recorded in the area managed by the "hard total" service, the application must also support "hard total" service.	Diebold/Dennis	I don't think this is required here. It is sufficient to allow access to the local hard disk or even remote files.	Eliminated the control from the description.	Changed Gesture Control Device must support the hard totals to might support the hard totals type description.		Resolved
Issue113	45 / Gesture Control	Device Sharing	Device Sharing The Gesture Control is an exclusive-use device, as follows:	Diebold/Dennis	We do not use "control" in UnifiedPOS, use device instead.	Eliminated the control from the description.	Changed the Gesture Control to Gesture Control Device.		Resolved
Issue114 Table229	45 / Gesture Control	Property	JointList Property Comma-separated list of joint information supported by the device. Each piece of joint information	Diebold/Dennis	There is only one line in the following table. What comes behind the colon?	Added the description in the JointID section.	Added the explanation into the JointList property parameter. And added the table to		Resolved

UPOS Ver1.16 RCSD Specification

			consists of the following information and is shown in the following order, separated by a colon (":").				explain those ones functions.		
Issue115 Table222 Table223	45 / Gesture Control	Property	AutoModelList Property For example, in conjunction with the camera, if the mode of tracking the face of a person by moving only the joint of Joint 01 and the mode of tracking by moving all joints are supported as follows. "FaceTrack_Joint 01, FaceTrack_ALL" (Content and order are dependent on the device.)	Diebold/ Dennis	Where are this values defined? What's there meaning?	Edited the description of Remarks. This is just an example to be used as mode. Therefore, all of the modes depend on the device. In the future this will be very popular but at this stage to make an concrete example is not easy.	Edited the AutoModelList Property Remarks. Added the table to explain those ones functions.		Resolved
Issue116 Table225 Table234	45 / Gesture Control	Property	CapMotion Property If true, the device supports pose function.	Diebold/ Dennis	What is the difference to the remark at property CapPose? What's the difference at all between both properties. Copy&Paste error?	Edited the description of Remarks.	Edited the CapMotion Property Remarks.		Resolved
Issue117 Table230	45 / Gesture Control	Property	MotionList Property Remarks Comma-separated list of motion IDs defined on the device.	Diebold/ Dennis	What values are in this list. Without them a standardization is meaningless, I think!	Added the several examples.	Edited the PoseList Property Remarks. And to make a standardization is not easy. This is the list of motion that is supported. It is up to the system.		Resolved
Issue118 Table226 Table227 Table231 Table235	45 / Gesture Control	Property	PoseList Property Remarks A comma-separated list of pause IDs defined on the device.	Diebold/ Dennis	What values are in this list. Without them a standardization is meaningless, I think!	Added the several examples.	Edited the MotionList Property Remarks.		Resolved
Issue119 Table218 Table236	45 / Gesture Control	Method	getPosition Method 1) getPosition (jointID: string, position: int32 by reference): 2)Parameter & Description jointID Specify the joint ID. Specify one of the values listed in the JointList property. However, it must be an ID whose position range exists or not. position The position of the joint specified by JointID is stored. 3) Remarks	Diebold/ Dennis	1)get position syntax is wrong as UnifiedPOS syntax. "out" parameter is required! 2) position range exists or not is unclear. 3) JointID is stored is unclear. 4) It acquires should be fetches.	Edited the jointID description.	Edited the getPosition Method description syntax and parameters. Also changed the parameter position:int32 by reference to out position:int32		Resolved

UPOS Ver1.16 RCSD Specification

			It acquires the position specified by jointID and stores it in position.						
Issue120 Table239	45 / Gesture Control	Method	startMotion Method Remarks Motion files need to be placed in the area managed by "hard total" service.	Diebold/Dennis	I don't think this is needed. Local HD is sufficient, I guess.	Changed the description of Hard Totals use from must to might.	Changed Gesture Control Device must support the hard totals to might support the hard totals type description.		Resolved
Issue121 Table240	45 / Gesture Control	Method	startPose Method Remarks This method is executed asynchronously.	Diebold/Dennis	Note about setting OutputID before returning is required here.	Added the description for this method about the OutputCompleteEvent to handle.	Added the description for this method about the OutputCompleteEvent to handle.		Resolved
Issue122 Table238 Table241	45 / Gesture Control	Method	stopControl Method	Diebold/Dennis	Event definitions are lacking!	Added the OutputCompleteEvent Event description.	Added the description for this method about the OutputCompleteEvent to handle.		Resolved
Issue123 Table250	46 / Device Monitor	Model	Model 1) *If the AutoDisable property is true, the device will automatically disable itself when a DataEvent is enqueued. 2) *In the device control, the measured value of the device is managed with an integer value of int32 type, but some devices handle decimal values. In that case, you can calculate the actual value by dividing the measured value by the factor for each device that can be acquired with the DeviceList property.	Diebold/Dennis	1) Not sure, whether this makes sense in context of monitoring. Please explain the use case. 2) We are not use "control" in UnifiedPOS spec, use "device" instead. "..." in that case the decimal places are implicit and the actual value must be calculated." It's called" coefficient later on.	Changed the description regarding the AutoDisable property. Also changed the items regarding the measure value and its calculation.	Edited the Device Monitor model description. Please keep in your mind this is just monitoring and store each device information in it.		Resolved
Issue124	46 / Device Monitor	Property	MonitoringDeviceList Property The values shown are as follows. "Device 01: 0: 0: 0: 0, Device 02: 1: 365: 0: 500"	Diebold/Dennis	Are whitespaces optional or mandatory?	All of the white spaces are eliminated.	All of the white spaces are eliminated.		Resolved
Issue125	47 / Graphic Display	General Info	General Information The Graphic Display has the following capability:	Diebold/Dennis	Could we maybe find a better wording here? Maybe "Static Display" or something – at the end static (non interactive or graphic) stuff is just displayed. Maybe "Projector" à native speaker	We would like to use the word Graphic Display for this device.	We would like to use the word Graphic Display for this device.		Resolved

UPOS Ver1.16 RCSD Specification

Issue126 Table266	47 / Graphic Display	Model	Image Display Mode Applications need to support "hard total" services as image files displaying with loadImage method must be placed in the area managed by the "hard total" service.	Diebold/ Dennis	Not sure, whether this is really required. Access to local HD is sufficient, I guess.	We think we need those 3 modes as Image Display, Video display and Web display. Hart Totals use will be not must but may.	We think we need those 3 modes as Image Display, Video display and Web display. Hart Totals use will be not must but may.		Resolved
Issue127 Table265 Table267	47 / Graphic Display	Model	Movie Display Mode => Video Display Mode	Diebold/ Dennis	This will be better Video not Movie.	Changed from movie to video	Changed from movie to video		Resolved
Issue128 Table268	47 / Graphic Display	Model	Web Display Mode The web display mode of the Graphics Display follows the general "Device Input Model" for event-driven input:	Diebold/ Dennis	I'm totally confused. Why is this an input model??? I would expect an output model. You just display the web page on the display, isn't it?	This is the Device Output Model not the device input model. Changed the description.	This is the Device Output Model not the device input model. Changed the description.		Resolved
Issue129 Table265	47 / Graphic Display	Device Sharing	Device Sharing The web browser is an exclusive-use device, as follows:	Diebold/ Dennis	This web browser should be a Graphic Display.	We changed the image display mode, video display mode and web display mode model descriptions.	Please refer to the revised model description for each mode.		Resolved
Issue130 Table282	47 / Graphic Display	Property	CapVideoTypeList Property Remarks Contains the comma-delimited list of video file type that are supported by the device. For example, if the device only supports AVI_IYUV and AVI_MJPG, then this property should be set to "AVI_IYUV,AVI_MJPG".	Diebold/ Dennis	Regarding the "AVI_IYUV and AVI_MJPG" Where this definitions come from? What's there meaning. Should not be device specific.	They are very common name as Video files.	They are very common name as Video files.		Resolved
Issue131 Table278	47 / Graphic Display	Property	LoadStatus Property 1) Syntax: LoadStatus: int32 {read-only, access after open} 2) Remarks Its value is set prior to a DataEvent being delivered to the application.	Diebold/ Dennis	1) Should be after open & enabled. 2) Model concern: OutputCompleteEvent is better here!	Changed after open-claim-enabled. Also, behavior of this property has been changed. It is prior to StatusUpdateEvent delivery specified values are set in the LoadStatus Property.	Changed after open-claim-enabled. Also, behavior of this property has been changed. It is prior to StatusUpdateEvent delivery specified values are set in the LoadStatus Property.		Resolved
Issue132 Table261 Table272 Table273 Table280	47 / Graphic Display	Property	URL, CapBack, CapForward Property 1) Syntax : URL: string {read-only, access after open} 2) Remarks	Diebold/ Dennis	1) Should be after open & enabled. 2)OutputCompleteEvent is definitely better here! See my	Changed the way to access this property from after open to after open claim enabled. Also defined to use the StatusUpdateEvent.	Changed the way to access this property from after open to after open claim enabled. Also defined to use the StatusUpdateEvent.		Resolved

UPOS Ver1.16 RCSD Specification

			Its value is set prior to a DataEvent being delivered to the application.		comment for web display mode. 3) Add the URL name in the property	Changed the CapBack, CapForward property name to CapURLBack, CapURLForward.	Also Changed the property name as CapURLBack and CapURLForward		
Issue133 Table286	47 / Graphic Display	Method	loadImage Method Remarks Image files must be located in the area managed by "Hard Total" service.	Diebold/ Dennis	Don't think this is really required. Access to local HD is sufficient.	Changed to use Hard Totals from must to might.	Changed Graphic Display Device loadImage Method must support the hard totals to might support the hard totals type description.		Resolved
Issue134 Table288	47 / Graphic Display	Method	1) playVideo Method 2) Remarks Video files must be located in the area managed by "Hard Total" service.	Diebold/ Dennis	1) How the application knows that a video is currently playing? Any property reports that? If so, it must be referred here. 2) Don't think this is really required. Access to local HD is sufficient.	Status will be reported by OutputCompleteEvent or ErrorEvent. Hard Total use description was changed from must to might.	Edited the playVideo Method Remarks and also changed Graphic Display device must support the Hard Totals to might support it. Added the StatusUpdateEvent then this concern will go away.		Resolved
Issue135 Table287	47 / Graphic Display	Method	loadURL Method Remarks The load status is reported by DataEvent	Diebold/ Dennis	OutputCompleteEvent is definitely better here! See my comment for web display mode.	Added the description regarding the OutputCompleteEvent.	Added the description regarding the OutputCompleteEvent.		Resolved
Issue136 Table263 Table284	47 / Graphic Display	Method	goBack Method Remarks The load status is reported by DataEvent	Diebold/ Dennis	OutputCompleteEvent is definitely better here! See my comment for web display mode. Also change the Method name from goBack to goURLBack.	Added the OutputCompleteEvent Event description. And changed the Method name as goURLBack	Added the description regarding the OutputCompleteEvent.		Resolved
Issue137 Table263 Table285	47 / Graphic Display	Method	goForward Method Remarks The load status is reported by DataEvent	Diebold/ Dennis	OutputCompleteEvent is definitely better here! See my comment for web display mode. Also change the Method name as goURLForward.	Added the OutputCompleteEvent Event description. And changed the Method name as goURLForward	Added the description regarding the OutputCompleteEvent.		Resolved
Issue138 Table263 Table290	47 / Graphic Display	Method	updatePage Method Remarks The load status is reported by DataEvent	Diebold/ Dennis	OutputCompleteEvent is definitely better here! See my comment for web display mode Also change the Method name as updateURLPage.	Added the OutputCompleteEvent Event description. And changed the Method name as updateURLPage	Added the description regarding the OutputCompleteEvent.		Resolved
Issue139 Table263 Table283	47 / Graphic Display	Method	cancelLoading Method 1) Remarks The load status is reported by DataEvent	Diebold/ Dennis	1) OutputCompleteEvent is definitely better here! See my comment for web display mode.	Added the OutputCompleteEvent Event description.	Added the description regarding the OutputCompleteEvent.		Resolved

UPOS Ver1.16 RCSD Specification

					2) Event definitions are lacking. 3) Change the Method name as cancelURLLoading.	Changed the Method name as cancelURLLoading			
Issue140 Table190 Table206	43 / Sound Player 44 / Speech Synthesis 45 / Gesture Control 47 / Graphic Display	Method	ClearOutput Method was missing even it is described in the chapter description	OPOS-J			Removed the not supported description		Resolved
Issue141 Table34 Table70 Table72 Table73 Table80 Table82 Table89 Table91 Table92 Table93 Table100 Table101 Table102 Table103 Table105 Table114 Table115 Table117 Table118 Table119 Table120 Table121 Table142 Table151 Table152 Table153 Table156 Table159 Table162 Table262 Table270	Property types CapXXX , XXX, XXXList 39/ VidoeCaptu re 41/ Sound Recorder 42/ Voice Recognition 43/ Sound Player 44/ Speech Synthesis 45/ Gesture Control 46/ Device Monitor 47/ Graphic Display	Property	Instead of using CapXXXList property only, it is preferring to use the combination of CapXXX, XXX and XXXList type of capabilities.	Diebold/ Dennis	It is historical UPOS way, but better to use for the application to handle in the future since it is simple.	OPOS-J would like to keep the current UPOS style about this. Therefore, we will change from CapXXXList type property to the CapXXX, XXX and XXXList type of capabilities and properties.	All of this related properties and capabilities will be changed this way.	Not Adopted	

UPOS Ver1.16 RCSD Specification

Table274 Table276 Table277 Table279 Table281 Table282									
Issue142	42/ Voice Recognition	Property	HearingDataPattern HearingDataWord HearingDataWordList =>Use the Hard instead of Hearing	Diebold/ Dennis	Hard is better than the Hearing	OPOS-J would like to keep the current UPOS style about this.		OPOS-J will use Hearing not Heard about this.	Not Adopted
Issue143	48/ Graphic Display	Event	DataEvent is not used but there is a description.	Diebold/ Dennis	DataEvent is not used but there is a description.	Eliminated the DataEvent from this chapter.	Eliminated the DataEvent from this chapter.		Resolved
Issue144 Table27 Table57 Table84 Table111 Table141 Table150 Table154 Table163 Table189 Table192 Table193 Table196 Table216 Table221 Table224 Table228 Table232 Table235 Table260 Table269 Table271	Use of HardTotals Description is not sufficient. 39/ Video Capture 42/ Sound Recorder 45/ Gesture Control 47/ Graphic Display	Property and Model description	Need to make clear the use of HardTotals and device behavior model description.	Diebold/ Dennis	For the HardTotals use it was decided to use those. They are CapAssociatedHardTotalsDevice, CapStorage and Storage. This was decided after the discussion with Retail DTF.	Made the decision use the CapAssociatedHardTotalsDevice CapStorage and Storage. Also added the E_EXTENDED Error. It is E_XXX_NOROOM since there is no error to tell the lack of storage data volume.	Please refer to the edited descriptions.		Resolved
Issue145 Table299	Elimination of DataEvent (Utilize the StatusUpdateEvent instead) 47/ Graphic Display	DataEvent related description		OPOS-J	For the device handling to use the StatusUpdate event will be better instead of DataEvent.	1. Changed the Device Model Description 2. Eliminated the DataEvent related properties. AutoDisable, DataCount, DataEventEnabled 3. Eliminated the DataEvent related method. clearInput, clearInputProperties			Resolved

UPOS Ver1.16 RCSD Specification

						4. Changed the ErrorEvent description.			
Issue146 Table178 Table275 Table286	Correction of incorrect abbreviation 42/Voice Recognition 44/ Speech Synthesis 47/Graphic Display		There are some incorrect device name abbreviations. t.	OPOS-J	1.Speech Synthesis abbreviation was described as SPSY. 2. Voice Recognition abbreviation was described as TTS 3. Graphic Display abbreviation was described as GDISP	Speech Synthesis abbreviation was corrected as SPCH. Voice Recognition abbreviation was corrected as VRCC. Graphic Display abbreviation was corrected as GDSP	OK		Resolved
Issue147	47/Graphic Display	Property and Class Diagram	ImageType Property type was incorrect. Video Type Property type was incorrect. Class diagram was incorrect.	OPOS-J	ImageType Property type was int32. VideoType Property type was int32. Graphic Display Class diagram had the DataEvent.	ImageType Property type was corrected as string. VideoType Property type was corrected as string. Eliminated the DataEvent from Graphic Display Class diagram.			Resolved
Issue148	45/Gesture Control	Method	createMotion Method Remarks	Diebold/Dennis	The place where the motion file is recorded is the area value in the Storage property is not a valid description.	Edited as follows. The place where the motion file is recorded is the area value of the Storage property.			Resolved
Issue149 Table7 Table17 Table37 Table43 Table130 Table145 Table171 Table188 Table204 Table205 Table215 Table217 Table246 Table247 Table259	All devices	Property, Method, Event	The use of Property, Method and Event was described Not Supported and/or Not supported. In some cases it was used but described Not supported	SEIKO EPSON/ Tad Furuhata	To make the uniformity it was decided to use Not supported. In some cases it was corrected from Not supported to exact version No.	Edited Not Supported to Not supported. In some cases Not supported to exact version number.			Resolved

UPOS Ver1.16 RCSD Specification

Issue150 Table8 Table194 Table195	All devices	May use after section of property.	There were several description regarding the May use after. It was not unified as UPOS specification.	SEIKO EPSON/ Tad Furuhata	To make the uniformity as UPOS spec, May use after description was changed.	May use after was changed from open, claim to open & claim.			Resolved.
Issue151 Table16 Table44 Table131 Table146 Table172 Table190 Table206 Table219 Table249 Table264	21/Lights 29/POS Power 39/Video Capture	Transition Event	It was missing	SEIKO EPSON/ Tad Furuhata	In the Event description Transition Event description was missing.	Added the Transition Event description			Resolved
Issue152 Table25 Table237	29/POS Power	Direct I/O, Model,...e.t.c.	There was a word of Control.	Diebold/ Dennis	UPOS should not use the wording of Control, or use the device control.	Edited the word of control, just elimination or use the word of device control.			Resolved
Issue153 Table128 Table200 Table201 Table202 Table203	39/Video Capture	Error Event	Error Response attribute was incorrect. Also, for the storage device use EXXX_NOROOM was added. EL_INPUT, EL_INPUT related description were eliminated since they are not used.	SEIKO EPSON/ Tad Furuhata	To make clear the Error Event behavior changed the description regarding the attribute and ErrorLocus .	Attribute and Error Extended and ErrorLocus Remarks section's descriptions were edited very much.			Resolved.
Issue154 Table137 Table138 Table139 Table140 Table147 Table166 Table167 Table168 Table169 Table173 Table184 Table185 Table186 Table187 Table211 Table212 Table213 Table214 Table242	All devices	Events	Some of the description of DataEvent, ErrorEvent, DirectIOEvent, OutputCompleteEvent and StatusUpdateEvent were incorrect.	SEIKO EPSON/ Tad Furuhata	Some Events need to add the description, some Events need to eliminate. Need to check based on the Model and Device behavior.	Need to add and eliminate the appropriated UPOS historical description.			Resolved

UPOS Ver1.16 RCSD Specification

Table243 Table244 Table245 Table256 Table257 Table258 Table291 Table292 Table293 Table294									
Issue155 Table248 Table254	46/Device Monitor	getDeviceValue Method	The getDeviceValue Method parameter description was incorrect.	SEIKO EPSON/ Tad Furuhata	Need to change the parameter.	Parameter was changed from inout value:int32 to pVaule:int32.			Resolved
Issue156 Table251 Table252 Table253	46/Device Monitor	addMonitoringDevice Method values were incorrect.	DMON_MMODE_HIGH DMON_MMODE_LOW DMON_MMODE_OUTSIDE descriptions	SEIKO EPSON/ Tad Furuhata	Need to change the description.	The description was corrected from we will notify the event each time to the event will be notified in each time.			Resolved
Issue157 Table121 Table289	47/Graphic Display	stopVideo Method description	Since Event handling was changed from DataEvent to StatusUpdateEvent and OutputCompleteEvent, need to change the description.	SEIKO EPSON/ Tad Furuhata	To make a quick Asynchronous device handling changed from DataEvent handling to StatusUpdateEvent and OutputCompleteEvent handling as a device	Description of DataEvent was eliminated and StatusUpdateEvent and OutputCompleteEvent handling description was added.			Resolved
Issue158 Table295 Table296 Table297 Table298	All devices	Method	clearInput clearInputProperties Method were missing.	SEIKO EPSON/ Tad Furuhata	Need to add them and need to change the common properties summary description.	Added the version number in the May use after section			Resolved
Issue159 Table300	46Device Monitor	Property	MonitoringDeviceList Property value was incorrect.	SEIKO EPSON/ Tad Furuhata	This property value included the white space like those. "Device01 : 0: 0: 0:0, Device02: 1: 365 :0 :500"	Now changed the property value as listed below. "Device01:0:0:0:0,Deveicie02 :1:365:0:500"			Resolved